WRDC-TR-90-8007
Volume VIII
Part 30

AD-A248 969

INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume VIII - User Interface Subsystem
Part 30 - Text Editor Product Specification

F. Glandorf

Control Data Corporation
Integration Technology Services
2970 Presidential Drive
Fairborn, OH 45324-6209

DTIC
ELECTE
APR 16 1992
S B D

September 1990

Final Report for Period 1 April 1987 - 31 December 1990

Approved for Public Release; Distribution is Unlimited

92-09612

MANUFACTURING TECHNOLOGY DIRECTORATE
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

92 4 14 047

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, regardless whether or not the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data. It should not, therefore, be construed or implied by any person, persons, or organization that the Government is licensing or conveying any rights or permission to manufacture, use, or market any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations

DAVID L. JUDSON, Project Manager
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

FOR THE COMMANDER:

BRUCE A. RASMUSSEN, Chief
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

If your address has changed, if you wish to be removed form our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/MTI, Wright-Patterson Air Force Base, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

FOREWORD

This technical report covers work performed under Air Force Contract F33600-87-C-0464, DAPro Project. This contract is sponsored by the Manufacturing Technology Directorate, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Bruce A. Rasmussen, Branch Chief, Integration Technology Division, Manufacturing Technology Directorate, through Mr. David L. Judson, Project Manager. The Prime Contractor was Integration Technology Services, Software Programs Division, of the Control Data Corporation, Dayton, Ohio, under the direction of Mr. W. A. Osborne. The DAPro Project Manager for Control Data Corporation was Mr. Jimmy P. Maxwell.

The DAPro project was created to continue the development, test, and demonstration of the Integrated Information Support System (IISS). The IISS technology work comprises enhancements to IISS software and the establishment and operation of IISS test bed hardware and communications for developers and users.

The following list names the Control Data Corporation subcontractors and their contributing activities:

SUBCONTRACTOR	ROLE
Control Data Corporation	Responsible for the overall Common Data Model design development and implementation, IISS integration and test, and technology transfer of IISS.
D. Appleton Company	Responsible for providing software information services for the Common Data Model and IDEF1X integration methodology.
ONTEK	Responsible for defining and testing a representative integrated system base in Artificial Intelligence techniques to establish fitness for use.
Simpact Corporation	Responsible for Communication development.
Structural Dynamics Research Corporation	Responsible for User Interfaces, Virtual Terminal Interface,and Network Transaction Manager design, development, implementation, and support.
Arizona State University	Responsible for test bed operations and support.

TABLE OF CONTENTS

		Page
SECTION 1.0	SCOPE	1-1
1.1	Identification	1-1
1.2	Functional Summary	1-1
SECTION 2.0	DOCUMENTS	2-1
2.1	Reference Documents	2-1
2.2	Terms and Abbreviations	2-2
SECTION 3.0	REQUIREMENTS	3-1
3.1	Structural Description	3-1
3.2	Functional Flow	3-1
3.3	Interfaces	3-2
3.3.1	Application Interface	3-3
3.3.2	UIM	3-3
3.3.3	NTM	3-3
3.4	Program Interrupts	3-3
3.5	Timing and Sequencing Description	3-3
3.6	Special Control Features	3-3
3.7	Storage Allocation	3-3
3.7.1	Data Base Definition	3-4
3.7.1.1	File Descriptions	3-4
3.8	Object Code Creation	3-4
3.9	Adaptation Data	3-4
3.10	Detailed Design Description	3-4
3.10.1	Main Program List	3-4
3.10.2	Module List	3-6
3.10.3	External Routines List	3-9
3.10.4	Include File List	3-11
3.10.5	Where Include File Used List	3-13
3.10.6	Where External Routine Used List	3-21
3.10.7	Main Program Parts List	3-29
3.10.8	Module Documentation	3-32
3.10.9	Include File Description	3-93
3.10.10	Hierarchy Chart	3-100
3.11	Program Listings Comments	3-121
SECTION 4.0	QUALITY ASSURANCE PROVISIONS	4-1
4.1	Introduction and Definitions	4-1
4.2	Computer Programming and Test Evaluation	4-1

LIST OF ILLUSTRATIONS

Figure	Title	Page
3-1	Text Editor Data Flow	3-2
3-2	Text Editor Interfaces	3-2

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

SECTION 1

SCOPE

1.1 Identification

This specification establishes the detailed design of a computer program identified as the Text Editor, hereinafter referred to as the TE. The TE is one configuration item of the Integrated Information Support System (IISS) User Interface (UI).

1.2 Functional Summary

The TE provides the user with the ability to edit a file in a manner consistent with the User Interface. Editing functions such as inserting, deleting, moving and replacing text are available.

SECTION 2

DOCUMENTS

2.1 Reference Documents

[1] Structural Dynamics Research Corporation, Application Interface Product Specification, PS 620144700 , 1 November 1985.

[2] Structural Dynamics Research Corporation, Forms Driven Form Editor Product Specification, PS 620144402 , 1 November 1985.

[3] Structural Dynamics Research Corporation, Forms Language Compiler Product Specification, PS 620144401 , 1 November 1985.

[4] Structural Dynamics Research Corporation, Form Processor Product Specification, PS 620144200 , 1 November 1985.

[5] Structural Dynamics Research Corporation, Rapid Application Generator Product Specification, PS 620144502 , 1 November 1985.

[6] Structural Dynamics Research Corporation, Report Writer Product Specification, PS 620144501 , 1 November 1985.

[7] Structural Dynamics Research Corporation, User Interface Services Product Specification, PS 620144100 , 1 November 1985.

[8] Structural Dynamics Research Corporation, Virtual Terminal Product Specification, PS 620144300 , 1 November 1985.

[9] Structural Dynamics Research Corporation, Text Editor Development Specification, DS 620144600B, 1 November 1985.

[10] Structural Dynamics Research Corporation, Text Editor User Manual, UM 620144600B, 1 November 1985.

[11] Structural Dynamics Research Corporation, Text Editor Unit Test Plan, UTP620144600 , 1 November 1985.

2.2 Terms and Abbreviations

Buffer Name: the default file in which the buffer will be saved if no file is given on a save command.

Current Cursor Position: the position of the cursor before an edit command or function is issued in the text editor.

Cursor Position: the position of the cursor after any command is issued.

Cut and Paste Buffer: where deleted lines go and the paste and fill edit commands get their data.

Display Start Line: the first line in the buffer to be displayed.

Display Size: the number of lines used in the edit area.

Field: two dimensional space on a terminal screen.

Field Pointer: indicates the ITEM which contains the current cursor position.

Form Processor: (FP), subset of the IISS User Interface that consists of a set of callable execution time routines available to an application program for form processing.

Form Processor Text Editor: (FPTE), subset of the Form Processor that consists of software modules that provide text editing capabilities to all users of applications that use the Form Processor.

IISS Function Screen: the first screen that is displayed after logon. It allows the user to specify the function he wants to access and the device type and device name on which he is working.

Integrated Information Support System: (IISS), a test computing environment used to investigate, demonstrate and test the concepts of information management and information integration in the context of Aerospace Manufacturing. The IISS addresses the problems of integration of data resident on heterogeneous data bases supported by heterogeneous computers interconnected via a Local Area Network.

Item: non-decomposable area of a form in which hard-coded descriptive text may be placed and the only defined areas where user data may be input/output.

Message: descriptive text which may be returned in the standard message line on the terminal screen. They are used to warn of errors or provide other user information.

Message Line: a line on the terminal screen that is used to display messgaes.

Operating System: (OS), software supplied with a computer which allows it to supervise its own operations and manage access to hardware facilities such as memory and peripherals.

Paging and Scrolling: a method which allows a form to contain more data than can be displayed with provisions for viewing any portion of the data buffer.

Previous Cursor Position: the position of the cursor when the previous edit command was issued.

Previous Edit Command: the function key pressed before the current one.

Select Line: one terminus of the select range.

Select Mode: when on, certain commands will be executed over the lines in the selected range. The commands are <DELETE LINE> and replace.

Text Editor: (TE), subset of the IISS User Interface that consists of a file editor that is based on the text editing functions built into the Form Processor.

Top of file: the first line of the buffer.

User Interface: (UI), IISS subsystem that controls the user's terminal and interfaces with the rest of the system. The UI consists of two major subsystems: the User Interface Development System (UIDS) and the User Interface Management System (UIMS).

User Interface Development System: (UIDS), collection of IISS User Interface subsystems that are used by applications programmers as they develop IISS applications. The UIDS includes the Form Editor and the Application Generator.

User Interface Services: (UIS), subset of the IISS User Interface that consists of a package of routines that aid users in controlling their environment. It includes message management, change password, and application definition services.

Virtual Terminal: (VT), subset of the IISS User Interface that performs the interfacing between different terminals and the UI. This is done by defining a specific set of terminal features and protocols which must be supported by the UI software which constitutes the virtual terminal definition. Specific terminals are then mapped against the virtual terminal software by specific software modules written for each type of real terminal supported.

SECTION 3

REQUIREMENTS

3.1 Structural Description

The Text Editor is an IISS application and as such can be invoked from the IISS Function Screen. It provides IISS users with file editing capabilities.

The Text Editor has two basic modes, edit and command. Edit mode is used most often and consists of entering text and using key commands called functions. The command mode is entered by pressing the <COMMAND> key in the edit mode. The command mode is exited by completing a command or by pressint the <QUIT> key.

Text files are input through the host operating system's file access method, modified by the TE's user, and then written back to the file. Deppending on the host this may overwrite the old version or create a new version of the file.

The detailed structure of the Text Editor is illustrated in section 3.10.

3.2 Functional Flow

Figure 3-1 is a data flow diagram of the Text Editor.

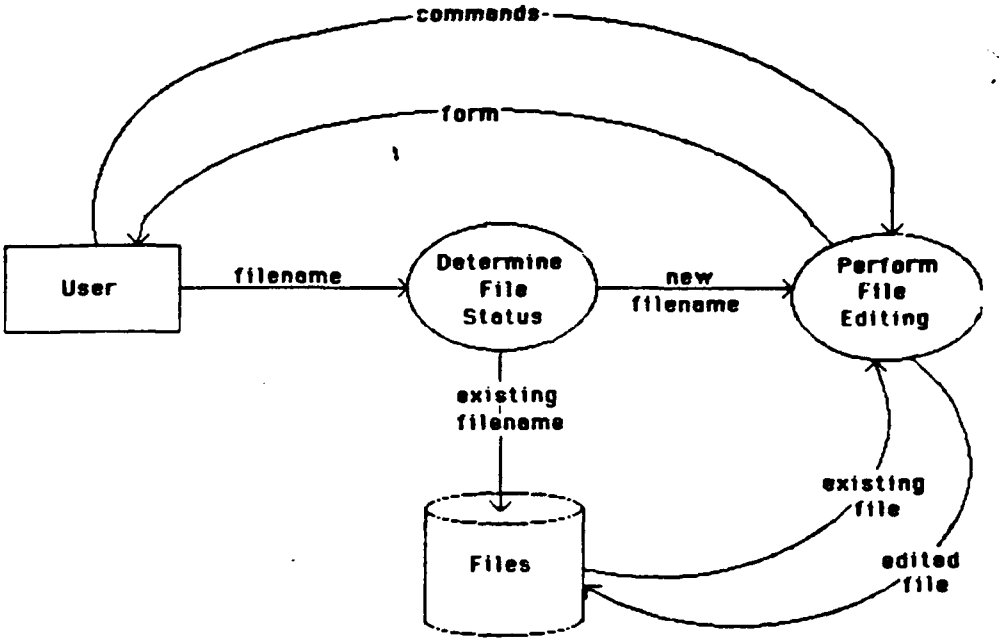


Figure 3-1 Text Editor Data Flow

3.3 Interfaces

Figure 3-2 describes the structure of the TE interfaces.

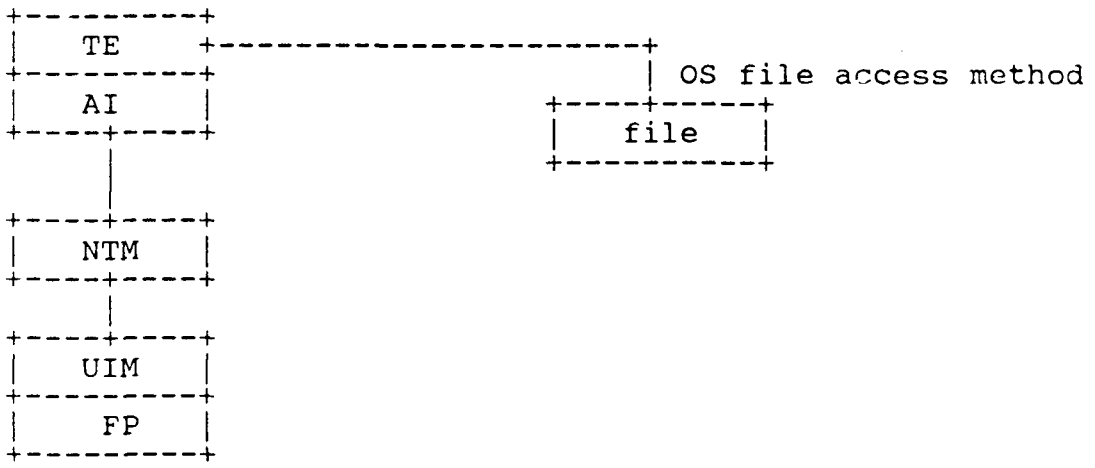


Figure 3-2 Text Editor Interfaces

3.3.1 Application Interface

The TE interfaces with the Form Processor through calls to the Application Interface (AI). The AI routines translate the application request into the NTM message format and then send the message that contains the application requiest data to the User Interface Monitor.

3.3.2 UIM

The UIM, which is the main control program for the Form Processor, calls the appropriate FP routine when a message is received from the Text Editor. The UIM then returns the output parameters to the TE.

3.3.3 NTM

The Text Editor application must also use the NTM initialization routine, INITAL and the NTM termination routine, TRMNAT.

3.4 Program Interrupts

This section does not apply to the detailed design of the Text Editor.

3.5 Timing and Sequencing Description

The TE is initiated through the IISS Function Screen presented to the user after correctly logging on to the IISS. The TE accepts input from and returns output to the user's terminal via defined forms. The TE permits the user to load files from host operating system and modify and save them.

3.6 Special Control Features

The detailed design of the Text Editor does not include any special control features as defined in the ICAM Documentation Standards manual.

3.7 Storage Allocation

The Text Editor executable is 163 blocks.

3.7.1 Data Base Definition

3.7.1.1 File Descriptions

1. FILE NAME: user specified

   PURPOSE: This is the file that is modified by the user of the Text Editor.

   DECLARATION:

   ```
   char line[79];
   ```

3.8 Object Code Creation

All TE modules were compiled using a C compiler developed by Interactive Software under VAX/VMS.

3.9 Adaptation Data

The C source modules for the Text Editor can be compiled using any UNIX version 7 compatible C compiler.

3.10 Detailed Design Description

3.10.1 Main Program List

The following is a list of all "Main Programs" which are modules that are not called by any other module being documented here. These modules are either program entry points or, if they are hooked into another set of programs via subroutine calls, they are the points the external programs can call and therefore enter through. To differentiate between the two types of entry points, look at the individual Module Documentation (section 3.10.8) and look at Module Type for each of the Main Program modules listed. Note whether the routine is a Program, Subroutine, or Function. If it is a Program, it is truly a main program entry point. If not, then it is merely called by other programs not being documented here.

TEXT EDITOR Main Program List

Module Name	Purpose
TE/MAIN	MAIN MODULE FOR TEXT EDITOR

3.10.2 Module List

The following is a list of all the modules being documented here along with their purpose. Each module has a unique name, no matter what language it was written in.

TEXT EDITOR Module List

Module Name	Purpose
CLEAR	CLEAR BUFFER
EDITCI	EDIT CALLABLE INTERFACE
EDITCI/ADDNODE	ADD NODE
EDITCI/CMDLIST	COMMAND LIST
EDITCI/CMDMENU	COMMAND MENU
EDITCI/COMMAND	COMMAND FUNCTION
EDITCI/CUT	CUT FUNCTION
EDITCI/DELLINE	DELETE LINE
EDITCI/DELNODE	DELETE NODE
EDITCI/DOCMD	DO A COMMAND
EDITCI/EDIT	EDIT LOOP
EDITCI/FILL	PASTE WITH FILL FUNCTION
EDITCI/FREENODE	FREE NODE
EDITCI/FRSTPAGE	FIRST PAGE
EDITCI/GETNODE	GET NODE
EDITCI/INDEX	FIND INDEX OF SUBSTRING
EDITCI/INLINE	INSERT LINE
EDITCI/LASTPAGE	LAST PAGE
EDITCI/MIDLINE	MIDLINE BREAK FUNCTION
EDITCI/NEWLINE	NEW LINE
EDITCI/NXTLINE	NEXT LINE

TEXT EDITOR Module List

Module Name	Purpose
EDITCI/NXTPAGE	NEXT PAGE
EDITCI/NXTVAL	NEXT VALUE OF COMMAND LINE
EDITCI/PFSRCH	CONTINUE SEARCH FUNCTION
EDITCI/PRVLINE	PREVIOUS LINE
EDITCI/PRVPAGE	PREVIOUS PAGE
EDITCI/PUTDEF	PUT DEFAULTS
EDITCI/QUIT	QUIT FUNCTION
EDITCI/RETRN	RETURN
EDITCI/SELECT	SELECT LINES FUNCTION
EDITCI/SPLIT	SPLIT A LINE
EDITCI/TMPCPY	TEMPORARY COPY
EDITCI/YANK	PASTE FUNCTION
LOAD	LOAD A FILE
MARGIN	SET MARGINS
NAME	NAME THE BUFFER
REPEAT	REPEAT FUNCTION/COMMAND
REPLACE	REPLACE COMMAND
SAVE	SAVE FILE
SEARCH	SEARCH COMMAND
TE/MAIN	MAIN MODULE FOR TEXT EDITOR

3.10.3 External Routines List

The following is a list of all routines or functions not documented here that are called by modules that are documented here. The first caller, in alphabetical order, is listed as well. See section 3.10.6 for a list of the modules that call each of these external routines.

TEXT EDITOR External Routines List

Module Name	First User
ABORT	EDITCI/INLINE
ADDFRM	EDITCI/CMDLIST
ATOI	REPEAT
ESCPY	EDITCI
FCLOSE	LOAD
FGETS	LOAD
FOPEN	LOAD
FWRITE	SAVE
GDATA	EDITCI/COMMAND
GETCUR	EDITCI/CMDLIST
GETMORE	EDITCI
GPAGE	EDITCI/CMDLIST
GWINDO	EDITCI/CMDLIST
INITAL	TE/MAIN
INITFP	TE/MAIN
ISPRINT	EDITCI/EDIT
MALLOC	EDITCI/GETNODE
MEMCMP	EDITCI/COMMAND
MEMCPY	EDITCI/CMDMENU
MEMSET	EDITCI/SELECT
OISCR	EDITCI/QUIT
PDATA	EDITCI/EDIT
PMSGLC	EDITCI/CMDLIST
PMSGLS	EDITCI/CMDLIST
PRINTF	EDITCI/INLINE
PUTATT	EDITCI/COMMAND
PUTCUR	EDITCI/COMMAND
PUTLOC	EDITCI/EDIT
RMVPAG	EDITCI/CMDMENU
STRCAT	REPLACE
STRCPY	EDITCI/PUTDEF
STRLEN	EDITCI
STRNCAT	REPLACE
STRNCMP	EDITCI/EDIT
STRUPC	EDITCI/DOCMD
TERMFP	TE/MAIN
TOUPPER	CLEAR
TRMNAT	TE/MAIN

3.10.4 Include File List

The following is a list of all include files called in by modules being documented here. Each include file has a unique name regardless of the language being used. The purpose of each include file is listed as well. A more complete description of each include file is given in section 3.10.9. The purpose listed is the one that is in the source code of the include file.

A purpose of "**** PURPOSE NOT FOUND BY STRIPPER ****" indicates that a purpose statement was not written into the include file itself. The most common reason for this is that the include file comes from system libraries that were not developed by the project, such as 'C' libraries that are provided with the 'C' compiler.

See section 3.10.6 for a set of lists which show all the modules which call in each of these include files.

TEXT EDITOR Include File List

File Name	Purpose
CTYPE	**** PURPOSE NOT FOUND BY STRIPPER ****
FPCODE	FORM PROCESSOR RETURN CODES
FPD	FORM PROCESSOR DATA
FPPARM	FORM PROCESSOR PARAMETERS
NTM	NTM INTERFACE INCLUDE FILE
STDIO	**** PURPOSE NOT FOUND BY STRIPPER ****
STDTYP	STANDARD TYPE DEFINITIONS

3.10.5 Where Include File Used List

The following lists each include file from 3.10.4 and all the modules documented in this specification which include them. The purpose of each module is listed as well.

TEXT EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
CTYPE		
	CLEAR	CLEAR BUFFER
	EDITCI	EDIT CALLABLE INTERFACE
	EDITCI/AD	ADD NODE
	EDITCI/CM	COMMAND LIST
	EDITCI/CM	COMMAND MENU
	EDITCI/CO	COMMAND FUNCTION
	EDITCI/CU	CUT FUNCTION
	EDITCI/DE	DELETE LINE
	EDITCI/DE	DELETE NODE
	EDITCI/DO	DO A COMMAND
	EDITCI/ED	EDIT LOOP
	EDITCI/FI	PASTE WITH FILL FUNCTION
	EDITCI/FR	FREE NODE
	EDITCI/FR	FIRST PAGE
	EDITCI/GE	GET NODE
	EDITCI/IN	FIND INDEX OF SUBSTRING
	EDITCI/IN	INSERT LINE
	EDITCI/LA	LAST PAGE
	EDITCI/MI	MIDLINE BREAK FUNCTION
	EDITCI/NE	NEW LINE
	EDITCI/NX	NEXT LINE
	EDITCI/NX	NEXT PAGE
	EDITCI/NX	NEXT VALUE OF COMMAND LINE
	EDITCI/PF	CONTINUE SEARCH FUNCTION
	EDITCI/PR	PREVIOUS LINE
	EDITCI/PR	PREVIOUS PAGE
	EDITCI/PU	PUT DEFAULTS
	EDITCI/QU	QUIT FUNCTION
	EDITCI/RE	RETURN
	EDITCI/SE	SELECT LINES FUNCTION
	EDITCI/SP	SPLIT A LINE
	EDITCI/TM	TEMPORARY COPY
	EDITCI/YA	PASTE FUNCTION
	LOAD	LOAD A FILE
	MARGIN	SET MARGINS
	NAME	NAME THE BUFFER
	REPEAT	REPEAT FUNCTION/COMMAND
	REPLACE	REPLACE COMMAND
	SAVE	SAVE FILE

TEXT EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
	SEARCH	SEARCH COMMAND
FPCODE		
	CLEAR	CLEAR BUFFER
	EDITCI	EDIT CALLABLE INTERFACE
	EDITCI/AD	ADD NODE
	EDITCI/CM	COMMAND LIST
	EDITCI/CM	COMMAND MENU
	EDITCI/CO	COMMAND FUNCTION
	EDITCI/CU	CUT FUNCTION
	EDITCI/DE	DELETE LINE
	EDITCI/DE	DELETE NODE
	EDITCI/DO	DO A COMMAND
	EDITCI/ED	EDIT LOOP
	EDITCI/FI	PASTE WITH FILL FUNCTION
	EDITCI/FR	FREE NODE
	EDITCI/FR	FIRST PAGE
	EDITCI/GE	GET NODE
	EDITCI/IN	FIND INDEX OF SUBSTRING
	EDITCI/IN	INSERT LINE
	EDITCI/LA	LAST PAGE
	EDITCI/MI	MIDLINE BREAK FUNCTION
	EDITCI/NE	NEW LINE
	EDITCI/NX	NEXT LINE
	EDITCI/NX	NEXT PAGE
	EDITCI/NX	NEXT VALUE OF COMMAND LINE
	EDITCI/PF	CONTINUE SEARCH FUNCTION
	EDITCI/PR	PREVIOUS LINE
	EDITCI/PR	PREVIOUS PAGE
	EDITCI/PU	PUT DEFAULTS
	EDITCI/QU	QUIT FUNCTION
	EDITCI/RE	RETURN
	EDITCI/SE	SELECT LINES FUNCTION
	EDITCI/SP	SPLIT A LINE
	EDITCI/TM	TEMPORARY COPY
	EDITCI/YA	PASTE FUNCTION
	LOAD	LOAD A FILE

TEXT EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
	MARGIN	SET MARGINS
	NAME	NAME THE BUFFER
	REPEAT	REPEAT FUNCTION/COMMAND
	REPLACE	REPLACE COMMAND
	SAVE	SAVE FILE
	SEARCH	SEARCH COMMAND
FPD		
	CLEAR	CLEAR BUFFER
	EDITCI	EDIT CALLABLE INTERFACE
	EDITCI/AD	ADD NODE
	EDITCI/CM	COMMAND LIST
	EDITCI/CM	COMMAND MENU
	EDITCI/CO	COMMAND FUNCTION
	EDITCI/CU	CUT FUNCTION
	EDITCI/DE	DELETE LINE
	EDITCI/DE	DELETE NODE
	EDITCI/DO	DO A COMMAND
	EDITCI/ED	EDIT LOOP
	EDITCI/FI	PASTE WITH FILL FUNCTION
	EDITCI/FR	FREE NODE
	EDITCI/FR	FIRST PAGE
	EDITCI/GE	GET NODE
	EDITCI/IN	FIND INDEX OF SUBSTRING
	EDITCI/IN	INSERT LINE
	EDITCI/LA	LAST PAGE
	EDITCI/MI	MIDLINE BREAK FUNCTION
	EDITCI/NE	NEW LINE
	EDITCI/NX	NEXT LINE
	EDITCI/NX	NEXT PAGE
	EDITCI/NX	NEXT VALUE OF COMMAND LINE
	EDITCI/PF	CONTINUE SEARCH FUNCTION
	EDITCI/PR	PREVIOUS LINE
	EDITCI/PR	PREVIOUS PAGE
	EDITCI/PU	PUT DEFAULTS
	EDITCI/QU	QUIT FUNCTION
	EDITCI/RE	RETURN

TEXT EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
	EDITCI/SE	SELECT LINES FUNCTION
	EDITCI/SP	SPLIT A LINE
	EDITCI/TM	TEMPORARY COPY
	EDITCI/YA	PASTE FUNCTION
	LOAD	LOAD A FILE
	MARGIN	SET MARGINS
	NAME	NAME THE BUFFER
	REPEAT	REPEAT FUNCTION/COMMAND
	REPLACE	REPLACE COMMAND
	SAVE	SAVE FILE
	SEARCH	SEARCH COMMAND
FPPARM		
	CLEAR	CLEAR BUFFER
	EDITCI	EDIT CALLABLE INTERFACE
	EDITCI/AD	ADD NODE
	EDITCI/CM	COMMAND LIST
	EDITCI/CM	COMMAND MENU
	EDITCI/CO	COMMAND FUNCTION
	EDITCI/CU	CUT FUNCTION
	EDITCI/DE	DELETE LINE
	EDITCI/DE	DELETE NODE
	EDITCI/DO	DO A COMMAND
	EDITCI/ED	EDIT LOOP
	EDITCI/FI	PASTE WITH FILL FUNCTION
	EDITCI/FR	FREE NODE
	EDITCI/FR	FIRST PAGE
	EDITCI/GE	GET NODE
	EDITCI/IN	FIND INDEX OF SUBSTRING
	EDITCI/IN	INSERT LINE
	EDITCI/LA	LAST PAGE
	EDITCI/MI	MIDLINE BREAK FUNCTION
	EDITCI/NE	NEW LINE
	EDITCI/NX	NEXT LINE
	EDITCI/NX	NEXT PAGE
	EDITCI/NX	NEXT VALUE OF COMMAND LINE
	EDITCI/PF	CONTINUE SEARCH FUNCTION

TEXT EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
	EDITCI/PR	PREVIOUS LINE
	EDITCI/PR	PREVIOUS PAGE
	EDITCI/PU	PUT DEFAULTS
	EDITCI/QU	QUIT FUNCTION
	EDITCI/RE	RETURN
	EDITCI/SE	SELECT LINES FUNCTION
	EDITCI/SP	SPLIT A LINE
	EDITCI/TM	TEMPORARY COPY
	EDITCI/YA	PASTE FUNCTION
	LOAD	LOAD A FILE
	MARGIN	SET MARGINS
	NAME	NAME THE BUFFER
	REPEAT	REPEAT FUNCTION/COMMAND
	REPLACE	REPLACE COMMAND
	SAVE	SAVE FILE
	SEARCH	SEARCH COMMAND
NTM		
	TE/MAIN	MAIN MODULE FOR TEXT EDITOR
STDIO		
	CLEAR	CLEAR BUFFER
	EDITCI	EDIT CALLABLE INTERFACE
	EDITCI/AD	ADD NODE
	EDITCI/CM	COMMAND LIST
	EDITCI/CM	COMMAND MENU
	EDITCI/CO	COMMAND FUNCTION
	EDITCI/CU	CUT FUNCTION
	EDITCI/DE	DELETE LINE
	EDITCI/DE	DELETE NODE
	EDITCI/DO	DO A COMMAND
	EDITCI/ED	EDIT LOOP
	EDITCI/FI	PASTE WITH FILL FUNCTION
	EDITCI/FR	FREE NODE

TEXT EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
	EDITCI/FR	FIRST PAGE
	EDITCI/GE	GET NODE
	EDITCI/IN	FIND INDEX OF SUBSTRING
	EDITCI/IN	INSERT LINE
	EDITCI/LA	LAST PAGE
	EDITCI/MI	MIDLINE BREAK FUNCTION
	EDITCI/NE	NEW LINE
	EDITCI/NX	NEXT LINE
	EDITCI/NX	NEXT PAGE
	EDITCI/NX	NEXT VALUE OF COMMAND LINE
	EDITCI/PF	CONTINUE SEARCH FUNCTION
	EDITCI/PR	PREVIOUS LINE
	EDITCI/PR	PREVIOUS PAGE
	EDITCI/PU	PUT DEFAULTS
	EDITCI/QU	QUIT FUNCTION
	EDITCI/RE	RETURN
	EDITCI/SE	SELECT LINES FUNCTION
	EDITCI/SP	SPLIT A LINE
	EDITCI/TM	TEMPORARY COPY
	EDITCI/YA	PASTE FUNCTION
	LOAD	LOAD A FILE
	MARGIN	SET MARGINS
	NAME	NAME THE BUFFER
	REPEAT	REPEAT FUNCTION/COMMAND
	REPLACE	REPLACE COMMAND
	SAVE	SAVE FILE
	SEARCH	SEARCH COMMAND

STDTYP

	CLEAR	CLEAR BUFFER
	EDITCI	EDIT CALLABLE INTERFACE
	EDITCI/AD	ADD NODE
	EDITCI/CM	COMMAND LIST
	EDITCI/CM	COMMAND MENU
	EDITCI/CO	COMMAND FUNCTION
	EDITCI/CU	CUT FUNCTION
	EDITCI/DE	DELETE LINE

TEXT EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
	EDITCI/DE	DELETE NODE
	EDITCI/DO	DO A COMMAND
	EDITCI/ED	EDIT LOOP
	EDITCI/FI	PASTE WITH FILL FUNCTION
	EDITCI/FR	FREE NODE
	EDITCI/FR	FIRST PAGE
	EDITCI/GE	GET NODE
	EDITCI/IN	FIND INDEX OF SUBSTRING
	EDITCI/IN	INSERT LINE
	EDITCI/LA	LAST PAGE
	EDITCI/MI	MIDLINE BREAK FUNCTION
	EDITCI/NE	NEW LINE
	EDITCI/NX	NEXT LINE
	EDITCI/NX	NEXT PAGE
	EDITCI/NX	NEXT VALUE OF COMMAND LINE
	EDITCI/PF	CONTINUE SEARCH FUNCTION
	EDITCI/PR	PREVIOUS LINE
	EDITCI/PR	PREVIOUS PAGE
	EDITCI/PU	PUT DEFAULTS
	EDITCI/QU	QUIT FUNCTION
	EDITCI/RE	RETURN
	EDITCI/SE	SELECT LINES FUNCTION
	EDITCI/SP	SPLIT A LINE
	EDITCI/TM	TEMPORARY COPY
	EDITCI/YA	PASTE FUNCTION
	LOAD	LOAD A FILE
	MARGIN	SET MARGINS
	NAME	NAME THE BUFFER
	REPEAT	REPEAT FUNCTION/COMMAND
	REPLACE	REPLACE COMMAND
	SAVE	SAVE FILE
	SEARCH	SEARCH COMMAND
	TE/MAIN	MAIN MODULE FOR TEXT EDITOR

3.10.6 Where External Routine Used List

The following lists each external function or routine listed in 3.10.3 and all the documented modules which call it. The purpose of each module is listed as well.

TEXT EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
ABORT		
	EDITCI/DEL	DELETE LINE
	EDITCI/INL	INSERT LINE
ADDFRM		
	EDITCI	EDIT CALLABLE INTERFACE
	EDITCI/CMD	COMMAND LIST
	EDITCI/CMD	COMMAND MENU
ATOI		
	MARGIN	SET MARGINS
	REPEAT	REPEAT FUNCTION/COMMAND
ESCPY		
	EDITCI	EDIT CALLABLE INTERFACE
FCLOSE		
	LOAD	LOAD A FILE
	SAVE	SAVE FILE
FGETS		
	LOAD	LOAD A FILE
FOPEN		
	LOAD	LOAD A FILE
	SAVE	SAVE FILE

TEXT EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
FWRITE		
	SAVE	SAVE FILE
GDATA		
	EDITCI/CMDCOMMAND LIST	
	EDITCI/CMDCOMMAND MENU	
	EDITCI/COMCOMMAND FUNCTION	
	EDITCI/EDIEDIT LOOP	
GETCUR		
	EDITCI/CMDCOMMAND LIST	
	EDITCI/EDIEDIT LOOP	
GETMORE		
	EDITCI	EDIT CALLABLE INTERFACE
GPAGE		
	EDITCI/CMDCOMMAND LIST	
GWINDO		
	EDITCI/CMDCOMMAND LIST	
INITAL		
	TE/MAIN	MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
INITFP		
	TE/MAIN	MAIN MODULE FOR TEXT EDITOR
ISPRINT		
	EDITCI/EDI	EDIT LOOP
MALLOC		
	EDITCI/GET	GET NODE
MEMCMP		
	EDITCI/CMD	COMMAND LIST
	EDITCI/COM	COMMAND FUNCTION
	EDITCI/EDI	EDIT LOOP
	TE/MAIN	MAIN MODULE FOR TEXT EDITOR
MEMCPY		
	EDITCI/CMD	COMMAND LIST
	EDITCI/CMD	COMMAND MENU
	EDITCI/EDI	EDIT LOOP
	EDITCI/INL	INSERT LINE
	NAME	NAME THE BUFFER
	REPLACE	REPLACE COMMAND
	SAVE	SAVE FILE
	SEARCH	SEARCH COMMAND
MEMSET		
	EDITCI	EDIT CALLABLE INTERFACE

TEXT EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
	EDITCI/CMDCOMMAND	MENU
	EDITCI/EDIEDIT	LOOP
	EDITCI/FILPASTE	WITH FILL FUNCTION
	EDITCI/PUTPUT	DEFAULTS
	EDITCI/SELSELECT	LINES FUNCTION
	LOAD	LOAD A FILE
OISCR		
	EDITCI/CMDCOMMAND	LIST
	EDITCI/CMDCOMMAND	MENU
	EDITCI/COMCOMMAND	FUNCTION
	EDITCI/EDIEDIT	LOOP
	EDITCI/QUIQUIT	FUNCTION
PDATA		
	EDITCI/CMDCOMMAND	MENU
	EDITCI/EDIEDIT	LOOP
	EDITCI/PUTPUT	DEFAULTS
PMSGLC		
	EDITCI/CMDCOMMAND	LIST
	EDITCI/CMDCOMMAND	MENU
	EDITCI/COMCOMMAND	FUNCTION
PMSGLS		
	EDITCI/CMDCOMMAND	LIST
	EDITCI/CMDCOMMAND	MENU
	EDITCI/COMCOMMAND	FUNCTION
	EDITCI/CUTCUT	FUNCTION
	EDITCI/DOCDO	A COMMAND
	EDITCI/EDIEDIT	LOOP
	EDITCI/FILPASTE	WITH FILL FUNCTION

TEXT EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
	EDITCI/FRSFIRST	PAGE
	EDITCI/INLINSERT	LINE
	EDITCI/LASLAST	PAGE
	EDITCI/NXTNEXT	LINE
	EDITCI/NXTNEXT	PAGE
	EDITCI/PRVPREVIOUS	LINE
	EDITCI/PRVPREVIOUS	PAGE
	EDITCI/QUIQUIT	FUNCTION
	REPEAT	REPEAT FUNCTION/COMMAND
PRINTF		
	EDITCI/DELDELETE	LINE
	EDITCI/EDIEDIT	LOOP
	EDITCI/INLINSERT	LINE
	TE/MAIN	MAIN MODULE FOR TEXT EDITOR
PUTATT		
	EDITCI/COMCOMMAND	FUNCTION
PUTCUR		
	EDITCI/COMCOMMAND	FUNCTION
	REPEAT	REPEAT FUNCTION/COMMAND
PUTLOC		
	EDITCI/EDIEDIT	LOOP
RMVPAG		
	EDITCI/CMDCOMMAND	LIST
	EDITCI/CMDCOMMAND	MENU

TEXT EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
STRCAT		
	EDITCI/PUTPUT	DEFAULTS
	REPLACE	REPLACE COMMAND
STRCPY		
	EDITCI	EDIT CALLABLE INTERFACE
	EDITCI/CMDCOMMAND	LIST
	EDITCI/CMDCOMMAND	MENU
	EDITCI/PFSCONTINUE	SEARCH FUNCTION
	EDITCI/PUTPUT	DEFAULTS
	EDITCI/SELSELECT	LINES FUNCTION
STRLEN		
	EDITCI	EDIT CALLABLE INTERFACE
	EDITCI/DOCDO	A COMMAND
	EDITCI/PUTPUT	DEFAULTS
	EDITCI/SELSELECT	LINES FUNCTION
	REPLACE	REPLACE COMMAND
	SEARCH	SEARCH COMMAND
STRNCAT		
	REPLACE	REPLACE COMMAND
STRNCMP		
	EDITCI	EDIT CALLABLE INTERFACE
	EDITCI/CMDCOMMAND	LIST
	EDITCI/CMDCOMMAND	MENU
	EDITCI/COMCOMMAND	FUNCTION
	EDITCI/DOCDO	A COMMAND

TEXT EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
	EDITCI/EDI	EDIT LOOP
	REPEAT	REPEAT FUNCTION/COMMAND
STRUPC		
	EDITCI/DOC	DO A COMMAND
TERMFP		
	TE/MAIN	MAIN MODULE FOR TEXT EDITOR
TOUPPER		
	CLEAR	CLEAR BUFFER
	EDITCI/IND	FIND INDEX OF SUBSTRING
TRMNAT		
	TE/MAIN	MAIN MODULE FOR TEXT EDITOR

3.10.7 Main Program Parts List

The following lists each Main Program listed in 3.10.1 and all the modules which are called either by that module itself or by any of the documented modules which it calls. It is possible for a non-main module to be listed more that once if it is called by multiple modules. The called modules, in this case known as program parts, are marked as to whether they are documented here. If so, the phrase "well-defined module" appears by the module name, if not it is an "external "routine". The Purpose of the Main Program module is listed as well.

TEXT EDITOR Main Program Parts List

Main Pgm Name	Module Name	Module Type
TE/MAIN	Purpose-->	MAIN MODULE FOR TEXT EDITOR
	ABORT	External routine
	ADDFRM	External routine
	ATOI	External routine
	CLEAR	Well-defined module
	EDITCI	Well-defined module
	EDITCI/ADDNODE	Well-defined module
	EDITCI/CMDLIST	Well-defined module
	EDITCI/CMDMENU	Well-defined module
	EDITCI/COMMAND	Well-defined module
	EDITCI/CUT	Well-defined module
	EDITCI/DELLINE	Well-defined module
	EDITCI/DELNODE	Well-defined module
	EDITCI/DOCMD	Well-defined module
	EDITCI/EDIT	Well-defined module
	EDITCI/FILL	Well-defined module
	EDITCI/FREENODE	Well-defined module
	EDITCI/FRSTPAGE	Well-defined module
	EDITCI/GETNODE	Well-defined module
	EDITCI/INDEX	Well-defined module
	EDITCI/INLINE	Well-defined module
	EDITCI/LASTPAGE	Well-defined module
	EDITCI/MIDLINE	Well-defined module
	EDITCI/NEWLINE	Well-defined module
	EDITCI/NXTLINE	Well-defined module
	EDITCI/NXTPAGE	Well-defined module
	EDITCI/NXTVAL	Well-defined module
	EDITCI/PFSRCH	Well-defined module
	EDITCI/PRVLINE	Well-defined module
	EDITCI/PRVPAGE	Well-defined module
	EDITCI/PUTDEF	Well-defined module
	EDITCI/QUIT	Well-defined module
	EDITCI/RETRN	Well-defined module
	EDITCI/SELECT	Well-defined module
	EDITCI/SPLIT	Well-defined module
	EDITCI/TMPCPY	Well-defined module
	EDITCI/YANK	Well-defined module
	ESCPY	External routine
	FCLOSE	External routine
	FGETS	External routine

TEXT EDITOR Main Program Parts List

Main Pgm Name	Module Name	Module Type
	FOPEN	External routine
	FWRITE	External routine
	GDATA	External routine
	GETCUR	External routine
	GETMORE	External routine
	GPAGE	External routine
	GWINDO	External routine
	INITAL	External routine
	INITFP	External routine
	ISPRINT	External routine
	LOAD	Well-defined module
	MALLOC	External routine
	MARGIN	Well-defined module
	MEMCMP	External routine
	MEMCPY	External routine
	MEMSET	External routine
	NAME	Well-defined module
	OISCR	External routine
	PDATA	External routine
	PMSGLC	External routine
	PMSGLS	External routine
	PRINTF	External routine
	PUTATT	External routine
	PUTCUR	External routine
	PUTLOC	External routine
	REPEAT	Well-defined module
	REPLACE	Well-defined module
	RMVPAG	External routine
	SAVE	Well-defined module
	SEARCH	Well-defined module
	STRCAT	External routine
	STRCPY	External routine
	STRLEN	External routine
	STRNCAT	External routine
	STRNCMP	External routine
	STRUPC	External routine
	TERMFP	External routine
	TOUPPER	External routine
	TRMNAT	External routine

3.10.8 Module Documentation

The following documentation describes information which is specific to each individual module being documented in this specification as listed in section 3.10.2. It provides a compact way of getting information that would be otherwise buried within each module's source code.

The specific items in this module documentation have the following meanings:

NAME:	Name of program Module.
PURPOSE:	Purpose of Module as detailed in the source code.
LANGUAGE:	Programming language source code is written in. The choices are: VAX-11 FORTRAN C (I/S-1 Workbench 'C') VAX-11 COBOL
MODULE TYPE:	Whether a Program, Subroutine, or Function.
SOURCE FILE:	Name of Source File from file specification.
SOURCE FILE TYPE:	Source File Extension from file specification.
HOST:	Whether this is a host-dependent routine (VAX or IBM) or blank if host-independent.
SUBSYSTEM:	IISS sub-system this file resides in.
SUBDIRECTORY:	Sub-directory of that subsystem in which this file resides.
DOCUMENTATION GROUP:	Name of documentation group of which this source file is a member.
DESCRIPTION:	A description of the module as otained from the source code.
ARGUMENTS:	The arguments with which this routine is called if it is a Subroutine or a Function.
INCLUDE FILES:	A list of all the files that are included into this module as well as their purposes.

ROUTINES CALLED:	Subroutines or Functions, either documented or external, called by this module, if any.
CALLED DIRECTLY BY:	The documented routines which call this module, if any.
USED IN MAIN PROGRAM(S):	The documented Main Programs which contain this module in their parts list according to the list in section 3.10.7.

The Module Documentation is arranged alphabetically according to Module Name.

TEXT EDITOR Module Documentation

```
NAME:                   CLEAR
PURPOSE:                CLEAR BUFFER
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  CHAR *CLEAR()

  DESCRIPTION - CLEAR

  IMPLEMENTS THE CLEAR BUFFER COMMAND.
  THE CONTENTS OF THE BUFFER SEL ARE REMOVED AND FREED AND THE
                      CONTENTS
  OF THE BUFFER BUF ARE MOVED TO THE SEL BUFFER. THE POINTERS
                      BUF_START
  AND CUR_ROW ARE MAINTAINED.
```

INCLUDE FILES:

STDTYP	- STANDARD TYPE DEFINITIONS
STDIO	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPD	- FORM PROCESSOR DATA
CTYPE	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPCODE	- FORM PROCESSOR RETURN CODES
FPPARM	- FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

EDITCI/ADDNODE - ADD NODE
EDITCI/DELLINE - DELETE LINE
EDITCI/FREENODE - FREE NODE
EDITCI/DELNODE - DELETE NODE
TOUPPER

CALLED DIRECTLY BY:

EDITCI/DOCMD - DO A COMMAND

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI
PURPOSE:                EDIT CALLABLE INTERFACE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          BOOL ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------

  SYNOPSIS
     BOOL EDITCI(FILE, CHANGED)
       CHAR FILE[];
       INT  *CHANGED;

     INPUTS:
       FILE - CHARACTER STRING WITH NAME OF FILE TO BE EDITED.
                    LENGTH
              IS 30. STRING MUST BE BLANK FILLED IN ALL
                    LANGUAGES EXCEPT C.

     OUTPUTS:
       CHANGED - POINTER TO FLAG INDICATING THAT THE FILE WAS
                    CHANGED.

  DESCRIPTION
  THIS STARTS UP THE EDITOR. IT IS THE ENTRY POINT IF CALLED
                    FROM A
  PROGRAM. RETURNS TRUE IF THE FILE HAS BEEN CHANGED ELSE
                    FALSE.
  INITIALIZES THE THREE BUFFERS, PUTS UP THE EDITING FORM, IF
                    SPECIFIED
  IT READS IN A FILE AND STARTS THE EDITING LOOP.

  DESCRIPTION

    FORMS PROCESSOR TEXT EDITOR. PROVIDES TEXT EDITING
                    CAPABILITY FOR
    IISS ENVIRONMENT.

    THE TE'S MODULE HIERARCHY IS RATHER FLAT. IN GENERAL THERE
                    IS ONE
    MODULE WHICH IMPLEMENTS EACH FUNCTION OR COMMAND. THESE
                    MODULES ARE
    JUST BELOW THE INITIALIZATION AND EDIT LOOP. UNDER THESE
                    ARE A SMALL
```

NUMBER OF ROUTINES WHICH PERFORM POINTER MAINTENANCE AND OTHER HOUSE
KEEPING FUNCTIONS.

PRIMARY DATA STRUCTURE CONSISTS OF THREE BUFFERS. THESE BUFFERS ARE
DOUBLY LINKED LISTS WITH A HEADER AND TRAILER NODE. THESE TWO NODES
ARE POINTED TO BY X_TOF AND X_BOF RESPECTIVELY, WHERE X IS THE NAME
OF THE BUFFER. THE NODE STRUCTURE IS DENOTED BY THE STRUCT
"TEXT_LINE". THE MAIN BUFFER CONTAINING THE FILE BEING EDITED HAS
SEVERAL OTHER POINTERS WHICH INDICATE THE FIRST LINE TO BE DISPLAYED,
THE LINE CONTAINING THE CURSOR AND A LINE WHICH IS A TERMINUS OF THE
SELECT RANGE.

BUF - BUFFER WHICH CONTAINS THE CONTENTS OF THE FILE BEING EDITED.
   BUF_TOF - A POINTER TO THE BUF'S HEADER NODE
   BUF_BOF - A POINTER TO THE BUF'S TRAILER NODE.
   CUR_ROW - A POINTER TO THE LINE CONTAINING THE CURSOR.
   BUF_START - A POINTER TO THE FIRST LINE OF BUF TO DISPLAY.
   INSEL - A POINTER TO THE LINE WHICH IS A TERMINUS OF THE SELECT
      RANGE. IT IS NULL IF NO SELECT RANGE IS ACTIVE.

SEL - BUFFER WHICH CONTAINS THE CONTENTS OF THE PASTE BUFFER.
   SEL_TOF - A POINTER TO THE SEL'S HEADER NODE.
   SEL_BOF - A POINTER TO THE SEL'S TRAILER NODE.

TMP - BUFFER FOR TEMPORARY HOLDING. LINES ARE INSERTED HERE FIRST
      THEN WHEN AN OPERATION IS SUCCESSFULLY COMPLETED THEY ARE MOVED
      TO BUFFER BUF.
   TMP_TOF - A POINTER TO THE TMP'S HEADER NODE.
   TMP_BOF - A POINTER TO THE TMP'S TRAILER NODE.

LINE NODES ARE ALLOCATED FROM A FREE POOL AND EXCEPT FOR INITIALIZATION
ALL NEXT AND PREVIOUS POINTER MAINTENANCE IS PERFORMED BY ONE OF FOUR
PROCEDURES.
   GETNODE - ALLOCATES A NODE FROM THE FREE LIST.
   FREENODE - RETURNS A NODE TO THE FREE LIST.
   ADDNODE - A GIVEN NODE IS LINKED INTO A BUFFER.
   DELNODE - REMOVES A GIVEN NODE FROM A BUFFER.

TWO HIGHER LEVEL PROCEDURES ARE USED TO INSERT STRINGS AND DELETE NODES
AND MAINTAIN THE POINTERS BUF_START AND CUR_ROW.
   INLINE - INSERTS THE GIVEN STRING INTO THE GIVEN

```
                    BUFFER. MAINTAINS
          THE POINTERS BUF_START AND CUR_ROW.
       DELLINE - REMOVES THE GIVEN NODE BUT DOES NOT FREE IT.
                    MAINTAINS
          THE POINTERS BUF_START AND CUR_ROW.

    THE POSITION THE CURSOR IS SUPPOSED TO BE AT IS KEPT IN
                    SCRN_ROW AND
    SCRN_COL. ALONG WITH BUF_START AND CUR_ROW THEY DETERMINE
                    WHICH
    PORTION OF THE BUF IS TO BE DISPLAYED AND THE CURSOR'S
                    POSITION.
    NEARLY ALL MODULES MANIPULATE THEIR VALUES.

ARGUMENTS:
----------
  FILE =        CHAR []
  CHANGED =        INT *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  GETMORE
  MEMSET
  ADDFRM
  STRNCMP
  ESCPY
  STRCPY
  EDITCI/DOCMD - DO A COMMAND
  EDITCI/FRSTPAGE - FIRST PAGE
  EDITCI/EDIT - EDIT LOOP
  EDITCI/DELNODE - DELETE NODE
  EDITCI/FREENODE - FREE NODE
  STRLEN


CALLED DIRECTLY BY:
-------------------
  TE/MAIN     - MAIN MODULE FOR TEXT EDITOR


USED IN MAIN PROGRAM(S):
------------------------
  TE/MAIN     - MAIN MODULE FOR TEXT EDITOR
```

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/ADDNODE
PURPOSE:                ADD NODE
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC VOID ADDNODE(P, Q)
     TEXT_LINE   *P, *Q;

  DESCRIPTION - ADDNODE

  INSERTS THE NODE POINTED TO BY P INTO THE LIST BEFORE THE
                    NODE POINTED
  TO BY Q.

ARGUMENTS:
----------
  P =         TEXT_LINE *
  Q =         TEXT_LINE *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


CALLED DIRECTLY BY:
-------------------
  EDITCI/CUT - CUT FUNCTION
  EDITCI/TMPCPY - TEMPORARY COPY
  CLEAR       - CLEAR BUFFER
  EDITCI/INLINE - INSERT LINE


USED IN MAIN PROGRAM(S):
------------------------
  TE/MAIN     - MAIN MODULE FOR TEXT EDITOR
```

TEXT EDITOR Module Documentation

NAME: EDITCI/CMDLIST
PURPOSE: COMMAND LIST
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: EDITCI
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: TE
DOCUMENTATION GROUP: TEXTEDT

DESCRIPTION:

STATIC INT CMDLIST()

DESCRIPTION - CMDLIST

IMPLEMENTS THE COMMAND LIST.
IT PERFORMS THE FOLLOWING:
1. DISPLAYS THE COMMAND LIST
2. GETS AND PARSES THE INPUT FROM THE COMMAND LIST.
3. GOES TO THE COMMAND'S FORM IF REQUIRED.
ELSE DOES THE COMMAND.
4. RETURNS THE FUNCTION KEY PRESSED.

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

GWINDO
STRNCMP
PMSGLC
RMVPAG
EDITCI/NXTVAL - NEXT VALUE OF COMMAND LINE
MEMCPY
STRCPY
EDITCI/DOCMD - DO A COMMAND
GETCUR
EDITCI/INDEX - FIND INDEX OF SUBSTRING
PMSGLS
EDITCI/CMDMENU - COMMAND MENU
GDATA
OISCR

MEMCMP
ADDFRM
GPAGE

CALLED DIRECTLY BY:

REPEAT - REPEAT FUNCTION/COMMAND
EDITCI/COMMAND - COMMAND FUNCTION

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/CMDMENU
PURPOSE:                COMMAND MENU
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          INT ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC INT CMDMENU(MENUNUM)
     INT MENUNUM;

  DESCRIPTION - CMDMENU

  IMPLEMENTS THE COMMAND FORM.
  IT PERFORMS THE FOLLOWING:
   1. DISPLAYS THE FORM WITH DEFAULTS FOR THE COMMAND
                      REQUESTED FROM THE
      COMMAND LIST (MENUNUM).
   2. GETS AND PARSES THE INPUT FROM THE COMMAND FORM.
   3. DOES THE COMMAND.
   4. RETURNS THE FUNCTION KEY PRESSED.

ARGUMENTS:
----------
  MENUNUM =         INT


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  ADDFRM
  STRNCMP
  PMSGLC
  RMVPAG
  EDITCI/NXTVAL - NEXT VALUE OF COMMAND LINE
  MEMCPY
  STRCPY
  EDITCI/DOCMD - DO A COMMAND
```

PMSGLS
MEMSET
GDATA
EDITCI/PUTDEF - PUT DEFAULTS
PDATA
OISCR

CALLED DIRECTLY BY:

EDITCI/CMDLIST - COMMAND LIST

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/COMMAND
PURPOSE:                COMMAND FUNCTION
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          INT ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC INT COMMAND()

  DESCRIPTION - COMMAND

  IMPLEMENTS THE COMMAND FUNCTION AND COMMAND LINE COMMAND.
  IT PERFORMS THE FOLLOWING:
   1. GUARDS THE EDIT AREA.
   2. GETS AND PARSES THE INPUT FROM THE COMMAND LINE.
   3. GOES TO THE COMMAND LINE IF REQUIRED.
      ELSE DOES THE COMMAND.
   4. RETURNS THE FUNCTION KEY PRESSED.


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  PUTATT
  STRNCMP
  PMSGLC
  MEMCMP
  EDITCI/CMDLIST - COMMAND LIST
  EDITCI/NXTVAL - NEXT VALUE OF COMMAND LINE
  EDITCI/DOCMD - DO A COMMAND
  PMSGLS
  GDATA
  OISCR
  PUTCUR
```

CALLED DIRECTLY BY:

EDITCI/EDI - EDIT LOOP
REPEAT - REPEAT FUNCTION/COMMAND

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/CUT
PURPOSE:                CUT FUNCTION
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC VOID CUT(DELCUT)
     BOOL   DELCUT;

  DESCRIPTION - CUT

  IMPLEMENTS THE DELETE LINE AND DELETE SELECT RANGE FUNCTION.
  REMOVES THE INDICATED NODE(S) AND PLACES IT(THEM) IN THE SEL
                        BUFFER.
  IF THE SELECT MODE IS ON THE SEL BUFFER IS CLEARED AND ALL
                        LINES IN
  THE SELECT RANGE ARE PUT IN THE SEL BUFFER.
  ELSE THE LINE WITH THE CURSOR IS PLACED IN THE SEL BUFFER.
  THE POINTERS BUF_START AND CUR_ROW ARE MAINTAINED TO
                        MINIMIZE THE
  VISUAL DISTURBANCE OF THE SCREEN.

ARGUMENTS:
----------
  DELCUT =        BOOL


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  EDITCI/SELECT - SELECT LINES FUNCTION
  EDITCI/SPLIT - SPLIT A LINE
  EDITCI/ADDNODE - ADD NODE
  EDITCI/MIDLINE - MIDLINE BREAK FUNCTION
```

EDITCI/FREENODE - FREE NODE
EDITCI/DELLINE - DELETE LINE
PMSGLS

CALLED DIRECTLY BY:

EDITCI/EDI - EDIT LOOP
REPEAT - REPEAT FUNCTION/COMMAND

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/DELLINE
PURPOSE:                DELETE LINE
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC VOID DELLINE(P)
     TEXT_LINE *P;

  DESCRIPTION - DELLINE

  REMOVES THE NODE POINTED TO BY P FROM ITS BUFFER BUT DOES
                      NOT FREE IT.
  THE POINTERS BUF_START AND CUR_ROW ARE MAINTAINED.

ARGUMENTS:
----------
  P =         TEXT_LINE *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  EDITCI/DELNODE - DELETE NODE
  ABORT
  PRINTF


CALLED DIRECTLY BY:
-------------------
  EDITCI/CUT - CUT FUNCTION
  EDITCI/SELECT - SELECT LINES FUNCTION
  CLEAR       - CLEAR BUFFER
  REPLACE     - REPLACE COMMAND
```

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

NAME: EDITCI/DELNODE
PURPOSE: DELETE NODE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: EDITCI
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: TE
DOCUMENTATION GROUP: TEXTEDT

DESCRIPTION:

```
STATIC VOID DELNODE(P)
   TEXT_LINE   *P;
```

DESCRIPTION - DELNODE

REMOVES THE NODE POINTED TO BY P FROM ITS BUFFER BUT DOES NO FREE IT.

ARGUMENTS:

P = TEXT_LINE *

INCLUDE FILES:

STDTYP	- STANDARD TYPE DEFINITIONS
STDIO	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPD	- FORM PROCESSOR DATA
CTYPE	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPCODE	- FORM PROCESSOR RETURN CODES
FPPARM	- FORM PROCESSOR PARAMETERS

CALLED DIRECTLY BY:

EDITCI/DELLINE	- DELETE LINE
EDITCI/TMPCPY	- TEMPORARY COPY
CLEAR	- CLEAR BUFFER
EDITCI	- EDIT CALLABLE INTERFACE
EDITCI/INLINE	- INSERT LINE

USED IN MAIN PROGRAM(S):

TE/MAIN	- MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/DOCMD
PURPOSE:                DO A COMMAND
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC CHAR *DOCMD()

  DESCRIPTION - DOCMD

  CALLS THE PROCEDURE IMPLEMENTING THE COMMAND CONTAINED IN
                      DATA[0].
  EACH PROCEDURE RETURNS A POINTER TO NULL IF SUCCESSFUL ELSE
                      THE
  POINTER IS TO A STRING WITH AN ERROR MESSAGE WHICH DOCMD
                      ISSUES.


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  NAME        - NAME THE BUFFER
  REPEAT      - REPEAT FUNCTION/COMMAND
  MARGIN      - SET MARGINS
  REPLACE     - REPLACE COMMAND
  CLEAR       - CLEAR BUFFER
  SAVE        - SAVE FILE
  SEARCH      - SEARCH COMMAND
  LOAD        - LOAD A FILE
  PMSGLS
  STRLEN
  STRNCMP
  STRUPC
```

CALLED DIRECTLY BY:

EDITCI/PFSRCH - CONTINUE SEARCH FUNCTION
REPEAT - REPEAT FUNCTION/COMMAND
EDITCI - EDIT CALLABLE INTERFACE
EDITCI/COMMAND - COMMAND FUNCTION
EDITCI/CMDLIST - COMMAND LIST
EDITCI/CMDMENU - COMMAND MENU

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/EDIT
PURPOSE:                EDIT LOOP
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC VOID EDIT()

  DESCRIPTION - EDIT

  PERFORMS THE EDIT LOOP:
   1. PUT DATA TO SCREEN.
   2. DO AN OISCR.
   3. GET DATA FROM SCREEN.
   4. CALL PRODURE IMPLEMENTING FUNCTION KEY PRESSED.
   5. REPEAT.


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  PMSGLS
  EDITCI/PFSRCH - CONTINUE SEARCH FUNCTION
  EDITCI/MIDLINE - MIDLINE BREAK FUNCTION
  EDITCI/CUT - CUT FUNCTION
  EDITCI/SELECT - SELECT LINES FUNCTION
  EDITCI/FILL - PASTE WITH FILL FUNCTION
  EDITCI/YANK - PASTE FUNCTION
  EDITCI/NEWLINE - NEW LINE
  EDITCI/COMMAND - COMMAND FUNCTION
  EDITCI/INDEX - FIND INDEX OF SUBSTRING
  EDITCI/LASTPAGE - LAST PAGE
  EDITCI/FRSTPAGE - FIRST PAGE
  EDITCI/PRVLINE - PREVIOUS LINE
  EDITCI/NXTLINE - NEXT LINE
  EDITCI/PRVPAGE - PREVIOUS PAGE
```

EDITCI/NXTPAGE - NEXT PAGE
EDITCI/QUIT - QUIT FUNCTION
GETCUR
MEMCMP
GDATA
OISCR
PUTLOC
STRNCMP
PDATA
PRINTF
ISPRINT
MEMSET
MEMCPY

CALLED DIRECTLY BY:

EDITCI - EDIT CALLABLE INTERFACE

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/FILL
PURPOSE:                PASTE WITH FILL FUNCTION
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC VOID FILL()

  DESCRIPTION - FILL

  IMPLEMENTS THE PASTE WITH FILL FUNCTION.
  INSERTS A COPY OF THE SEL BUFFER JUST BEFORE THE CUR_ROW
                    ACCORDING TO THE
  CURRENT FILL MARGINS (FILMRGN.LEFT AND FILMRGN.RIGHT)


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  EDITCI/TMPCPY - TEMPORARY COPY
  PMSGLS
  EDITCI/INLINE - INSERT LINE
  MEMSET


CALLED DIRECTLY BY:
-------------------
  EDITCI/EDI  - EDIT LOOP
  REPEAT      - REPEAT FUNCTION/COMMAND


USED IN MAIN PROGRAM(S):
------------------------
  TE/MAIN     - MAIN MODULE FOR TEXT EDITOR
```

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/FREENODE
PURPOSE:                FREE NODE
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC VOID FREENODE(P)
     TEXT_LINE   *P;

  DESCRIPTION - FREENODE

  PUTS THE NODE POINTED TO BY P BACK ON THE FREE LIST.

ARGUMENTS:
----------
  P =         TEXT_LINE *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


CALLED DIRECTLY BY:
-------------------
  EDITCI/CUT - CUT FUNCTION
  EDITCI/SELECT - SELECT LINES FUNCTION
  CLEAR       - CLEAR BUFFER
  REPLACE     - REPLACE COMMAND
  EDITCI      - EDIT CALLABLE INTERFACE
  EDITCI/INLINE - INSERT LINE


USED IN MAIN PROGRAM(S):
------------------------
  TE/MAIN     - MAIN MODULE FOR TEXT EDITOR
```

TEXT EDITOR Module Documentation

NAME: EDITCI/FRSTPAGE
PURPOSE: FIRST PAGE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: EDITCI
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: TE
DOCUMENTATION GROUP: TEXTEDT

DESCRIPTION:

STATIC VOID FRSTPAGE()

DESCRIPTION - FRSTPAGE

IMPLEMENTS THE FIRST PAGE FUNCTION.
SETS BUF_START SO THE FIRST DSP_HT LINES OF BUF ARE DISPLAYED.

INCLUDE FILES:

STDTYP	- STANDARD TYPE DEFINITIONS
STDIO	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPD	- FORM PROCESSOR DATA
CTYPE	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPCODE	- FORM PROCESSOR RETURN CODES
FPPARM	- FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

PMSGLS

CALLED DIRECTLY BY:

EDITCI/EDI	- EDIT LOOP
REPLACE	- REPLACE COMMAND
EDITCI	- EDIT CALLABLE INTERFACE

USED IN MAIN PROGRAM(S):

TE/MAIN	- MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/GETNODE
PURPOSE:                GET NODE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          TEXT_LINE * ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC TEXT_LINE *GETNODE()

  DESCRIPTION - GETNODE

  GETS A TEXT_LINE NODE FROM THE FREE LIST AND RETURNS A
                        POINTER TO IT.
  NODES ARE ALLOCATED 100 AT A TIME. IF GETNODE FAILS IT
                        RETURNS A NULL.


INCLUDE FILES:
--------------
  STDTYP        - STANDARD TYPE DEFINITIONS
  STDIO         - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD           - FORM PROCESSOR DATA
  CTYPE         - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE        - FORM PROCESSOR RETURN CODES
  FPPARM        - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  MALLOC


CALLED DIRECTLY BY:
-------------------
  EDITCI/INLINE - INSERT LINE


USED IN MAIN PROGRAM(S):
------------------------
  TE/MAIN       - MAIN MODULE FOR TEXT EDITOR
```

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/INDEX
PURPOSE:                FIND INDEX OF SUBSTRING
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          INT ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  SYNOPSIS
     INT INDEX(STR, SUB)
     CHAR *STR, *SUB;

  DESCRIPTION
     INDEX RETURNS THE OFFSET OF THE SUBSTRING IN THE STRING
                     OR -1 IF THE
     STRING DOES NOT CONTAIN THE SUBSTRING. THE UPPER CASE
                     REPRESENTATION
     OF THE STRINGS ARE USED.

ARGUMENTS:
----------
  STR =         CHAR *
  SUB =         CHAR *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  TOUPPER


CALLED DIRECTLY BY:
-------------------
  EDITCI/EDI  - EDIT LOOP
  SEARCH      - SEARCH COMMAND
  REPLACE     - REPLACE COMMAND
  EDITCI/CMDLIST - COMMAND LIST
  EDITCI/PUTDEF - PUT DEFAULTS
```

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/INLINE
PURPOSE:                INSERT LINE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          BOOL ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC BOOL INLINE(P, LINE)
     TEXT_LINE   *P;
     CHAR   LINE[];

  DESCRIPTION - INLINE

  GETS A NODE PUTS THE STRING POINTED TO BY LINE INTO IT AND
                        INSERTS IT
  INTO THE LIST JUST BEFORE THE NODE POINTED TO BY P. THE
                        POINTER
  BUF_START IS MAINTAINED.

  IF INLINE IS UNABLE TO OBTAIN A NODE IT ISSUES A MESSAGE AND
                        CLEARS
  THE TMP BUFFER.

ARGUMENTS:
----------
  P =         TEXT_LINE *
  LINE =         CHAR []


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  PRINTF
  ABORT
  EDITCI/GETNODE - GET NODE
  PMSGLS
  EDITCI/DELNODE - DELETE NODE
```

EDITCI/FREENODE - FREE NODE
MEMCPY
EDITCI/ADDNODE - ADD NODE

CALLED DIRECTLY BY:

EDITCI/NEWLINE - NEW LINE
EDITCI/YAN - PASTE FUNCTION
EDITCI/FIL - PASTE WITH FILL FUNCTION
EDITCI/SELECT - SELECT LINES FUNCTION
EDITCI/SPLIT - SPLIT A LINE
LOAD - LOAD A FILE
REPLACE - REPLACE COMMAND

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/LASTPAGE
PURPOSE:                LAST PAGE
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC VOID LASTPAGE()

  DESCRIPTION - LASTPAGE

  IMPLEMENTS THE LAST PAGE FUNCTION.
  SETS BUF_START SO THE LAST DSP_HT LINES OF BUF ARE DISPLAYED.


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  PMSGLS


CALLED DIRECTLY BY:
-------------------
  EDITCI/EDI  - EDIT LOOP
  REPLACE     - REPLACE COMMAND


USED IN MAIN PROGRAM(S):
------------------------
  TE/MAIN     - MAIN MODULE FOR TEXT EDITOR
```

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/MIDLINE
PURPOSE:                MIDLINE BREAK FUNCTION
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC VOID MIDLINE()

  DESCRIPTION - MIDLINE

  IMPLEMENTS THE MIDLINE BREAK FUNCTION.
  SPLITS THE LINE POINTED TO BY CUR_ROW AT SCRN_COL AND SETS
                      THE CURSOR TO
  THE BEGINNING OF THE NEW LINE.


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  EDITCI/RETRN - RETURN
  EDITCI/SPLIT - SPLIT A LINE


CALLED DIRECTLY BY:
-------------------
  EDITCI/EDI - EDIT LOOP
  EDITCI/CUT - CUT FUNCTION
  EDITCI/SELECT - SELECT LINES FUNCTION
  REPEAT      - REPEAT FUNCTION/COMMAND


USED IN MAIN PROGRAM(S):
------------------------
  TE/MAIN     - MAIN MODULE FOR TEXT EDITOR
```

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/NEWLINE
PURPOSE:                NEW LINE
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC VOID NEWLINE()

  DESCRIPTION - NEWLINE

  IMPLEMENTS THE INSERT LINE FUNCTION.
  INSERTS A BLANK LINE AT CUR_ROW.


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  EDITCI/INLINE - INSERT LINE


CALLED DIRECTLY BY:
-------------------
  EDITCI/EDI  - EDIT LOOP
  REPEAT      - REPEAT FUNCTION/COMMAND


USED IN MAIN PROGRAM(S):
------------------------
  TE/MAIN     - MAIN MODULE FOR TEXT EDITOR
```

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/NXTLINE
PURPOSE:                NEXT LINE
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC VOID NXTLINE()

  DESCRIPTION - NXTLINE

  IMPLEMENTS THE NEXT LINE FUNCTION.
  ADVANCES BUF_START BY ONE LINE TO THE NEXT LINE IS DISPLAYED.


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  PMSGLS


CALLED DIRECTLY BY:
-------------------
  EDITCI/EDI  - EDIT LOOP
  REPEAT      - REPEAT FUNCTION/COMMAND


USED IN MAIN PROGRAM(S):
------------------------
  TE/MAIN     - MAIN MODULE FOR TEXT EDITOR
```

TEXT EDITOR Module Documentation

NAME: EDITCI/NXTPAGE
PURPOSE: NEXT PAGE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: EDITCI
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: TE
DOCUMENTATION GROUP: TEXTEDT

DESCRIPTION:

STATIC VOID NXTPAGE()

DESCRIPTION - NXTPAGE

IMPLEMENTS THE NEXT PAGE FUNCTION.
ADVANCES BUF_START SO THE NEXT DSP_HT LINES OF THE BUFFER ARE DISPLAYED.

INCLUDE FILES:

STDTYP	- STANDARD TYPE DEFINITIONS
STDIO	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPD	- FORM PROCESSOR DATA
CTYPE	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPCODE	- FORM PROCESSOR RETURN CODES
FPPARM	- FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

PMSGLS

CALLED DIRECTLY BY:

EDITCI/EDI	- EDIT LOOP
REPEAT	- REPEAT FUNCTION/COMMAND

USED IN MAIN PROGRAM(S):

TE/MAIN	- MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/NXTVAL
PURPOSE:                NEXT VALUE OF COMMAND LINE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          INT ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC INT NXTVAL(CMD_LINE)
     CHAR   CMD_LINE[];

  DESCRIPTION - NXTVAL

  PARSES THE CMD_LINE ARGUMENT AND LEAVES EACH OPERAND IN A
                      DATA[I].
  EACH OPERAND IS '\0' TERMINATED. PUTS THE NUMBER OF
                      ARGUMENTS FOUND
  IN NUMDATA.

ARGUMENTS:
----------
  CMD_LINE =        CHAR []


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


CALLED DIRECTLY BY:
-------------------
  EDITCI/COMMAND - COMMAND FUNCTION
  EDITCI/CMDLIST - COMMAND LIST
  EDITCI/CMDMENU - COMMAND MENU


USED IN MAIN PROGRAM(S):
------------------------
  TE/MAIN     - MAIN MODULE FOR TEXT EDITOR
```

TEXT EDITOR Module Documentation

```
NAME.                   EDITCI/PFSRCH
PURPOSE:                CONTINUE SEARCH FUNCTION
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC VOID PFSRCH()

  DESCRIPTION - PFSRCH

  IMPLEMENTS THE FIND NEXT OCCURRENCE OF THE STRING AS A
                      FUNCTION.


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  EDITCI/DOCMD - DO A COMMAND
  STRCPY


CALLED DIRECTLY BY:
-------------------
  EDITCI/EDI  - EDIT LOOP
  REPEAT      - REPEAT FUNCTION/COMMAND


USED IN MAIN PROGRAM(S):
------------------------
  TE/MAIN     - MAIN MODULE FOR TEXT EDITOR
```

TEXT EDITOR Module Documentation

NAME: EDITCI/PRVLINE
PURPOSE: PREVIOUS LINE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: EDITCI
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: TE
DOCUMENTATION GROUP: TEXTEDT

DESCRIPTION:

STATIC VOID PRVLINE()

DESCRIPTION - PRVLINE

IMPLEMENTS THE PREVIOUS LINE FUNCTION.
BACKS UP THE BUF_START SO THE PREVIOUS LINE OF BUF IS DISPLAYED.

INCLUDE FILES:

STDTYP	- STANDARD TYPE DEFINITIONS
STDIO	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPD	- FORM PROCESSOR DATA
CTYPE	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPCODE	- FORM PROCESSOR RETURN CODES
FPPARM	- FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

PMSGLS

CALLED DIRECTLY BY:

EDITCI/EDI	- EDIT LOOP
REPEAT	- REPEAT FUNCTION/COMMAND

USED IN MAIN PROGRAM(S):

TE/MAIN	- MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/PRVPAGE
PURPOSE:                PREVIOUS PAGE
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC VOID PRVPAGE()

  DESCRIPTION - PRVPAGE

  IMPLEMENTS THE PREVIOUS PAGE FUNCTION.
  BACKS UP BUF_START SO THE PREVIOUS DSP_HT LINES OF THE
                     BUFFER ARE
  DISPLAYED.


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  PMSGLS


CALLED DIRECTLY BY:
-------------------
  EDITCI/EDI  - EDIT LOOP
  REPEAT      - REPEAT FUNCTION/COMMAND


USED IN MAIN PROGRAM(S):
------------------------
  TE/MAIN     - MAIN MODULE FOR TEXT EDITOR
```

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/PUTDEF
PURPOSE:                PUT DEFAULTS
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          INT ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC PUTDEF(FIELD, DATA)
     CHAR FIELD[], DATA[];

  DESCRIPTION - PUTDEF

  PUTS DEFAULT DATA IN THE FIELD.

ARGUMENTS:
----------
  FIELD =        CHAR []
  DATA =        CHAR []


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  EDITCI/INDEX - FIND INDEX OF SUBSTRING
  STRCPY
  STRLEN
  MEMSET
  PDATA
  STRCAT


CALLED DIRECTLY BY:
-------------------
  EDITCI/CMDMENU - COMMAND MENU
```

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

NAME: EDITCI/QUIT
PURPOSE: QUIT FUNCTION
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: BOOL ()
SOURCE FILE: EDITCI
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: TE
DOCUMENTATION GROUP: TEXTEDT

DESCRIPTION:

STATIC BOOL QUIT()

DESCRIPTION - QUIT

IMPLEMENTS THE QUIT KEY FUNCTION.
IF THE BUFFER HAS BEEN MODIFIED (SINCE THE INITIAL LOAD OR THE LAST SAVE)
YOU MUST PRESS THE QUIT KEY TWICE. PRESSING ANY OTHER KEY ABORTS THE QUIT
AND RETURNS TO EDIT.

IF THE BUFFER HAS NOT BEEN MODIFIED THE PROGRAM TERMINATES.

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

PMSGLS
OISCR

CALLED DIRECTLY BY:

EDITCI/EDI - EDIT LOOP

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/RETRN
PURPOSE:                RETURN
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC VOID RETRN()

  DESCRIPTION - RETRN

  SETS THE CURSOR TO THE BEGINNING OF THE NEXT LINE. THE
                        POINTERS BUF_START
  AND CUR_ROW ARE MAINTAINED.


INCLUDE FILES:
--------------
  STDTYP        - STANDARD TYPE DEFINITIONS
  STDIO         - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD           - FORM PROCESSOR DATA
  CTYPE         - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE        - FORM PROCESSOR RETURN CODES
  FPPARM        - FORM PROCESSOR PARAMETERS


CALLED DIRECTLY BY:
-------------------
  EDITCI/MIDLINE - MIDLINE BREAK FUNCTION


USED IN MAIN PROGRAM(S):
------------------------
  TE/MAIN       - MAIN MODULE FOR TEXT EDITOR
```

TEXT EDITOR Module Documentation

```
NAME:                   EDITCI/SELECT
PURPOSE:                SELECT LINES FUNCTION
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  STATIC VOID SELECT()

  DESCRIPTION - SELECT

  IMPLEMENTS THE MARK SELECT RANGE FUNCTION.
  IF THE SELECT RANGE IS OFF, IT IS TURNED ON AND THE CUR_ROW
                      IS MARKED
  AS ONE TERMINUS OF THE SELECT RANGE.
  ELSE THE SELECT RANGE IS TURNED OFF.


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  EDITCI/FREENODE - FREE NODE
  EDITCI/DELLINE - DELETE LINE
  EDITCI/INLINE - INSERT LINE
  STRLEN
  STRCPY
  MEMSET
  EDITCI/MIDLINE - MIDLINE BREAK FUNCTION


CALLED DIRECTLY BY:
-------------------
  EDITCI/EDI - EDIT LOOP
  EDITCI/CUT - CUT FUNCTION
  REPLACE    - REPLACE COMMAND
```

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

NAME: EDITCI/SPLIT
PURPOSE: SPLIT A LINE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: EDITCI
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: TE
DOCUMENTATION GROUP: TEXTEDT

DESCRIPTION:

STATIC VOID SPLIT()

DESCRIPTION - SPLIT

SPLITS THE LINE POINTED TO BY CUR_ROW AT SCRN_COL.

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

EDITCI/INLINE - INSERT LINE

CALLED DIRECTLY BY:

EDITCI/CUT - CUT FUNCTION
EDITCI/MIDLINE - MIDLINE BREAK FUNCTION

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

NAME: EDITCI/TMPCPY
PURPOSE: TEMPORARY COPY
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: EDITCI
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: TE
DOCUMENTATION GROUP: TEXTEDT

DESCRIPTION:

```
STATIC VOID TMPCPY(MOVE_START)
   BOOL MOVE_START;
```

DESCRIPTION - TMPCPY

MOVES THE CONTENTS OF THE TMP BUFFER TO THE BUF BUFFER JUST BEFORE THE CUR_ROW. THE POINTER BUF_START IS MAINTAINED IF MOVE_START IS TRUE.

ARGUMENTS:

MOVE_START = BOOL

INCLUDE FILES:

STDTYP	- STANDARD TYPE DEFINITIONS
STDIO	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPD	- FORM PROCESSOR DATA
CTYPE	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPCODE	- FORM PROCESSOR RETURN CODES
FPPARM	- FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

EDITCI/ADDNODE - ADD NODE
EDITCI/DELNODE - DELETE NODE

CALLED DIRECTLY BY:

EDITCI/YAN	- PASTE FUNCTION
EDITCI/FIL	- PASTE WITH FILL FUNCTION
LOAD	- LOAD A FILE
REPLACE	- REPLACE COMMAND

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

NAME: EDITCI/YANK
PURPOSE: PASTE FUNCTION
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: EDITCI
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: TE
DOCUMENTATION GROUP: TEXTEDT

DESCRIPTION:

STATIC VOID YANK()

DESCRIPTION - YANK

IMPLEMENTS THE PASTE FUNCTION.
INSERTS A COPY OF THE PASTE BUFFER BEFORE CUR_ROW

INCLUDE FILES:

STDTYP	- STANDARD TYPE DEFINITIONS
STDIO	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPD	- FORM PROCESSOR DATA
CTYPE	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPCODE	- FORM PROCESSOR RETURN CODES
FPPARM	- FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

EDITCI/TMPCPY - TEMPORARY COPY
EDITCI/INLINE - INSERT LINE

CALLED DIRECTLY BY:

EDITCI/EDI - EDIT LOOP
REPEAT - REPEAT FUNCTION/COMMAND

USED IN MAIN PROGRAM(S):

TE_MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   LOAD
PURPOSE:                LOAD A FILE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  CHAR *LOAD()

  DESCRIPTION - LOAD

  IMPLEMENTS THE LOAD FILE COMMAND.
  LOADS THE FILE NAMED BY DATA[1] AND INSERTS IT BEFORE
                      CUR_ROW IN THE BUFFER
  BUF.


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  FCLOSE
  EDITCI/TMPCPY - TEMPORARY COPY
  EDITCI/INLINE - INSERT LINE
  FGETS
  MEMSET
  FOPEN


CALLED DIRECTLY BY:
-------------------
  EDITCI/DOCMD - DO A COMMAND


USED IN MAIN PROGRAM(S):
------------------------
  TE/MAIN     - MAIN MODULE FOR TEXT EDITOR
```

TEXT EDITOR Module Documentation

NAME: MARGIN
PURPOSE: SET MARGINS
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: EDITCI
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: TE
DOCUMENTATION GROUP: TEXTEDT

DESCRIPTION:

CHAR *MARGIN()

DESCRIPTION - MARGIN

IMPLEMENTS THE SET FILL MARGINS COMMAND.
SETS THE LEFT MARGIN TO DATA[1] AND THE RIGHT MARGIN TO DATA[2].

INCLUDE FILES:

STDTYP	- STANDARD TYPE DEFINITIONS
STDIO	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPD	- FORM PROCESSOR DATA
CTYPE	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPCODE	- FORM PROCESSOR RETURN CODES
FPPARM	- FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

ATOI

CALLED DIRECTLY BY:

EDITCI/DOCMD - DO A COMMAND

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   NAME
PURPOSE:                NAME THE BUFFER
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  CHAR *NAME()

  DESCRIPTION - NAME

  IMPLEMENTS THE GIVE THE BUFFER A DEFAULT SAVE NAME.
  SETS THE DEFAULT SAVE NAME TO DATA[1].


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  MEMCPY


CALLED DIRECTLY BY:
-------------------
  EDITCI/DOCMD - DO A COMMAND


USED IN MAIN PROGRAM(S):
------------------------
  TE/MAIN     - MAIN MODULE FOR TEXT EDITOR
```

TEXT EDITOR Module Documentation

```
NAME:                   REPEAT
PURPOSE:                REPEAT FUNCTION/COMMAND
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  CHAR *REPEAT()

  DESCRIPTION - REPEAT

  IMPLEMENTS THE REPEAT COMMAND/FUNCTION COMMAND.
  THE COMMAND OR FUNCTION WHICH FOLLOWS WILL BE PERFORMED THE
                    NUMBER
  OF TIMES GIVEN BY DATA[1].


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  EDITCI/PFSRCH - CONTINUE SEARCH FUNCTION
  EDITCI/MIDLINE - MIDLINE BREAK FUNCTION
  EDITCI/CUT - CUT FUNCTION
  EDITCI/FILL - PASTE WITH FILL FUNCTION
  EDITCI/YANK - PASTE FUNCTION
  EDITCI/NEWLINE - NEW LINE
  EDITCI/PRVLINE - PREVIOUS LINE
  EDITCI/NXTLINE - NEXT LINE
  EDITCI/PRVPAGE - PREVIOUS PAGE
  EDITCI/NXTPAGE - NEXT PAGE
  EDITCI/DOCMD - DO A COMMAND
  EDITCI/COMMAND - COMMAND FUNCTION
  EDITCI/CMDLIST - COMMAND LIST
  STRNCMP
  PUTCUR
  PMSGLS
  ATOI
```

CALLED DIRECTLY BY:

EDITCI/DOCMD -- DO A COMMAND

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   REPLACE
PURPOSE:                REPLACE COMMAND
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  CHAR *REPLACE()

  DESCRIPTION - REPLACE

  IMPLEMENTS THE REPLACE STRING COMMAND.
  THE STRING DATA[1] IS REPLACED BY STRING DATA[2] ACCORDING
                     TO THE OPTION
  IN DATA[3] (OR THE SELECT RANGE IF ACTIVE). THESE BECOME THE
                     DEFAULTS
  FROM_STR, TO_STR, AND REPTYP RESPECTIVELY. IF DATA[1..3] ARE
                     UNSPECIFIED
  THE DEFAULTS ARE USED. THE POINTERS CUR_ROW AND BUF_START
                     ARE MAINTAINED.
  WARNING: IF REPLACE CANNOT INSERT A LINE THE REPLACEMENTS
                     PERFORMED ARE
  NOT BACKED OUT BUT THE CURSOR IS LEFT AT THE LINE WHERE THE
                     FAILURE
  OCCURED. NO LINE IS LEFT PARTIALLY REPLACED.


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  EDITCI/SELECT - SELECT LINES FUNCTION
  EDITCI/LASTPAGE - LAST PAGE
  EDITCI/FRSTPAGE - FIRST PAGE
  EDITCI/FREENODE - FREE NODE
  EDITCI/DELLINE - DELETE LINE
  EDITCI/TMPCPY - TEMPORARY COPY
  EDITCI/INLINE - INSERT LINE
```

STRCAT
STRNCAT
STRLEN
EDITCI/INDEX - FIND INDEX OF SUBSTRING
MEMCPY

CALLED DIRECTLY BY:

EDITCI/DOCMD - DO A COMMAND

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   SAVE
PURPOSE:                SAVE FILE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  CHAR *SAVE()

  DESCRIPTION - SAVE

  IMPLEMENTS THE WRITE BUFFER TO A FILE COMMAND.
  WRITES THE BUFFER BUF TO THE FILE NAMED BY DATA[1]. IF NO
                        NAME IS
  SPECIFIED THE NAME IN WORK_NAME IS USED.


INCLUDE FILES:
--------------
  STDTYP        - STANDARD TYPE DEFINITIONS
  STDIO         - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD           - FORM PROCESSOR DATA
  CTYPE         - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE        - FORM PROCESSOR RETURN CODES
  FPPARM        - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  FCLOSE
  FWRITE
  MEMCPY
  FOPEN


CALLED DIRECTLY BY:
-------------------
  EDITCI/DOCMD - DO A COMMAND


USED IN MAIN PROGRAM(S):
------------------------
  TE/MAIN       - MAIN MODULE FOR TEXT EDITOR
```

TEXT EDITOR Module Documentation

```
NAME:                   SEARCH
PURPOSE:                SEARCH COMMAND
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            EDITCI
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------
  CHAR *SEARCH()

  DESCRIPTION - SEARCH

  IMPLEMENTS THE SEARCH BUFFER FOR A STRING.
  THE BUFFER BUF IS SEARCHED FOR THE STRING CONTAINED IN
                        DATA[1] IN THE
  DIRECTION SPECIFIED IN DATA[2]. THESE BECOME THE DEFAULT
                        SEARCH VALUES
  SRCH_LINE AND SRCHDIR RESPECTIVELY. THE SEARCH IS STARTED
                        FROM CUR_ROW
  AND SCRN_COL. IF DATA[1] AND DATA[2] ARE UNSPECIFIED THEN
                        SRCH_LINE AND
  SRCHDIR ARE USED RESPECTIVELY. THE POINTERS BUF_START AND
                        CUR_ROW ARE
  MAINTAINED.


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS


ROUTINES CALLED:
----------------
  EDITCI/INDEX - FIND INDEX OF SUBSTRING
  STRLEN
  MEMCPY


CALLED DIRECTLY BY:
-------------------
  EDITCI/DOCMD - DO A COMMAND
```

USED IN MAIN PROGRAM(S):

TE/MAIN - MAIN MODULE FOR TEXT EDITOR

TEXT EDITOR Module Documentation

```
NAME:                   TE/MAIN
PURPOSE:                MAIN MODULE FOR TEXT EDITOR
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            TE
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           TE
DOCUMENTATION GROUP:    TEXTEDT

DESCRIPTION:
------------

  SYNOPSIS
     VOID MAIN(ARGC, ARGV)
       INT ARGC;
       CHAR *ARGV[];

   INPUTS:
          ARGC - ARGUMENT COUNT
          ARCV - ARGUMENT VECTOR

  DESCRIPTION - MAIN

  CALLS EDITCI WITH A FILE NAME, IF SPECIFIED ELSE CALLS IT
                    WITH NULL
  STRING.


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  NTM         - NTM INTERFACE INCLUDE FILE


ROUTINES CALLED:
----------------
  INITAL
  MEMCMP
  PRINTF
  INITFP
  EDITCI      - EDIT CALLABLE INTERFACE
  TERMFP
  TRMNAT
```

3.10.9 Include File Descriptions

The following list contains a purpose and description of each include file listed in 3.10.4 as specified in the source code. The language it is written in is also given.

TEXT EDITOR Include File Description

FILE NAME: FPCODE
PURPOSE: FORM PROCESSOR RETURN CODES
LANGUAGE: C

DESCRIPTION:

TEXT EDITOR Include File Description

FILE NAME: FPD
PURPOSE: FORM PROCESSOR DATA
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
DATA DEFINITIONS FOR ALL FORM PROCESSOR (INCLUDING MONITER)DATA.

TEXT EDITOR Include File Description

FILE NAME: FPPARM
PURPOSE: FORM PROCESSOR PARAMETERS
LANGUAGE: C

DESCRIPTION:

DESCRIPTION: THESE DATA DEFINITIONS ARE USED
IN THE FORM PROCESSOR ROUTINES.

TEXT EDITOR Include File Description

FILE NAME: NTM
PURPOSE: NTM INTERFACE INCLUDE FILE
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
INCLUDE FILE FOR NTM INTERFACE

TEXT EDITOR Include File Description

FILE NAME: STDTYP
PURPOSE: STANDARD TYPE DEFINITIONS
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
THIS FILE ENSURES THAT THE FOLLOWING STANDARD TYPES ARE AVAILABLE:

FLOAT	- SINGLE PRECISION FLOAT
DOUBLE	- DOUBLE PRECISION FLOAT
LONG	- 32 BIT (OR LARGER) SIGNED INTEGER
LBITS	- 32 BITS (OR MORE) FOR BIT MANIPULATION
INT	- NATURAL SIZE SIGNED INTEGER
UNSIGNED	- NATURAL SIZE UNSIGNED INTEGER
BOOL	- NATURAL SIZE LOGICAL (ZERO / NON-ZERO ONLY)
SHORT	- 16 BIT (OR LARGER) SIGNED INTEGER
USHORT	- 16 BIT (OR LARGER) UNSIGNED INTEGER
BITS	- 16 BITS (OR MORE) FOR BIT MANIPULATION
CHAR	- SINGLE MACHINE CHARACTER (REAL CHARACTERS ALWAYS POSITIVE)
TINY	- 8 BIT (OR LARGER) SIGNED INTEGER
UTINY	- 8 BIT (OR LARGER) UNSIGNED INTEGER
TBITS	- 8 BITS (OR MORE) FOR BIT MANIPULATION
TBOOL	- 8 BIT (OR LARGER) LOGICAL (ZERO / NON-ZERO ONLY)
METACHAR	- 16 BIT (OR LARGER) AUGMENTED CHARACTER (SIGNED)
VOID	- FUNCTION THAT RETURNS NO VALUE
FORTRAN	- STORAGE CLASS FOR FOREIGN (NON-C) ROUTINES OR C ROUTINES WHICH ARE CALLABLE FROM FOREIGN ROUTINES

SINCE NOT ALL COMPILERS SUPPORT USHORT, TINY, AND UTINY, THE FUNCTIONS
USHORT(), TINY(), AND UTINY() SHOULD BE USED WHENEVER REFERENCING THEM.

IN ADDITION, THE FOLLOWING UTILITY MACROS ARE DEFINED:

LURSHIFT(N, B)	- UNSIGNED LONG RIGHT SHIFT
MAX(A, B)	- MAXIMUM OF A AND B
MIN(A, B)	- MINIMUM OF A AND B

TEXT EDITOR Include File Description

```
ABS(A)          - ABSOLUTE VALUE OF A
STRASN(A, B)    - TRANSPORTABLE A = B FOR STRUCTURES
NULL            - NULL POINTER VALUE (0)
TRUE            - 1
FALSE           - 0
SUCCESS         - EXIT(SUCCESS) INDICATES SUCCESSFUL
                COMPLETION
FAILURE         - EXIT(FAILURE) INDICATES ERRORS
```

THE FOLLOWING SYMBOLS SHOULD BE DEFINED BASED ON THE COMPILER BEING USED:

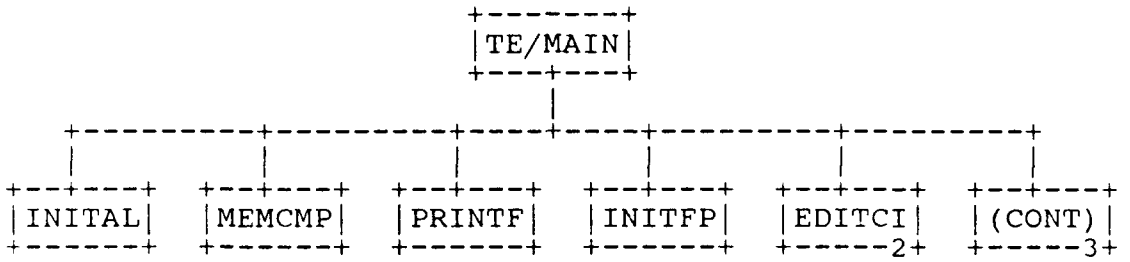
```
USHORT  - COMPILER SUPPORTS UNSIGNED SHORT
TINY    - COMPILER TREATS CHAR AS SIGNED
UTINY   - CHAR IS SIGNED AND COMPILER SUPPORTS
            UNSIGNED CHAR
VOID    - COMPILER SUPPORTS VOID
FORTRAN - COMPILER SUPPORTS FORTRAN
STRASN  - DEFINE APPROPRIATE MACRO
SUCCESS - DEFINE APPROPRIATE VALUE IF NOT 0
FAILURE - DEFINE APPROPRIATE VALUE IF NOT 1
```

3.10.10 Hierarchy Chart

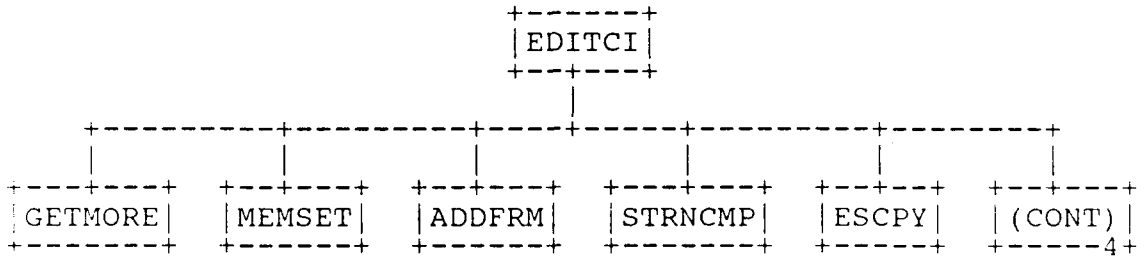
The following hierarchy charts show the relationships between all of the modules mentioned in the above documentation. A module may call a subroutine several times within its code, but the call will only be shown once as a single relationship on this hierarchy chart. All modules shown at the top of the first page are considered Main Programs as described in section 3.10.1 above.

There is an internal paging scheme as marked by the numbers in the upper right corner of each page. An index after the last page of the chart shows where a routine and its calls are first defined. If a routine has no page reference, it either makes no calls or is an external routine. A continuation box on the end of a tree limb shows where that the tree continues on the page numbered mentioned. A number in a box with a routine name points to the page where the routine is further defined within the hierarchy tree. If there is no number in a box, the routine either makes no calls or is an external routine.

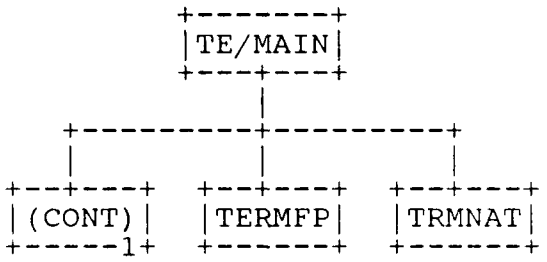
1


TE/MAIN
INITAL
MEMCMP
PRINTF
INITFP
EDITCI
(CONT)
2
3

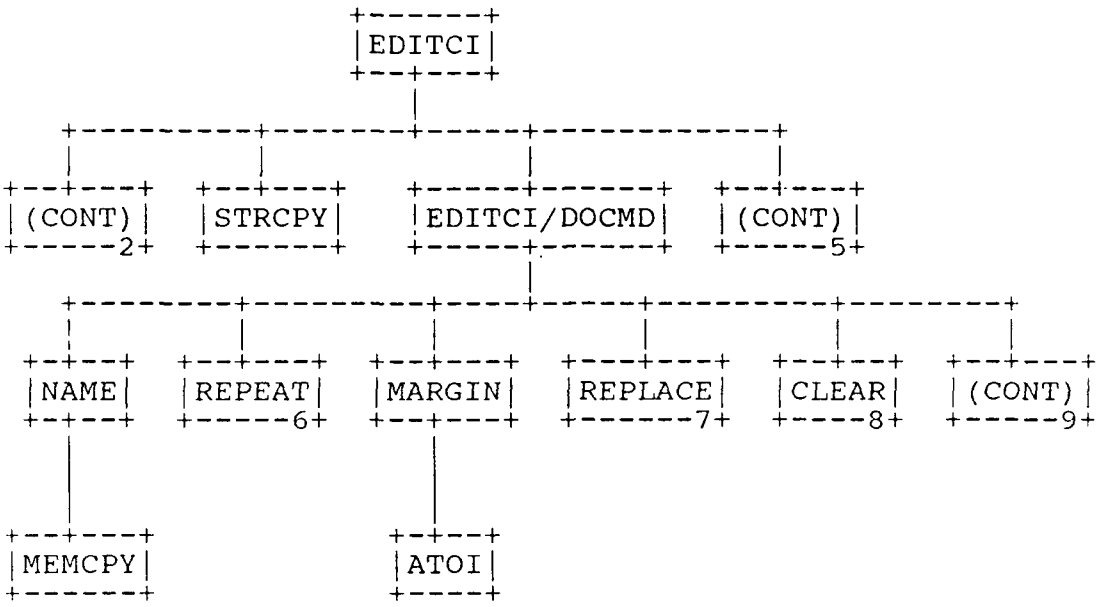
2


EDITCI
GETMORE
MEMSET
ADDFRM
STRNCMP
ESCPY
(CONT)
4

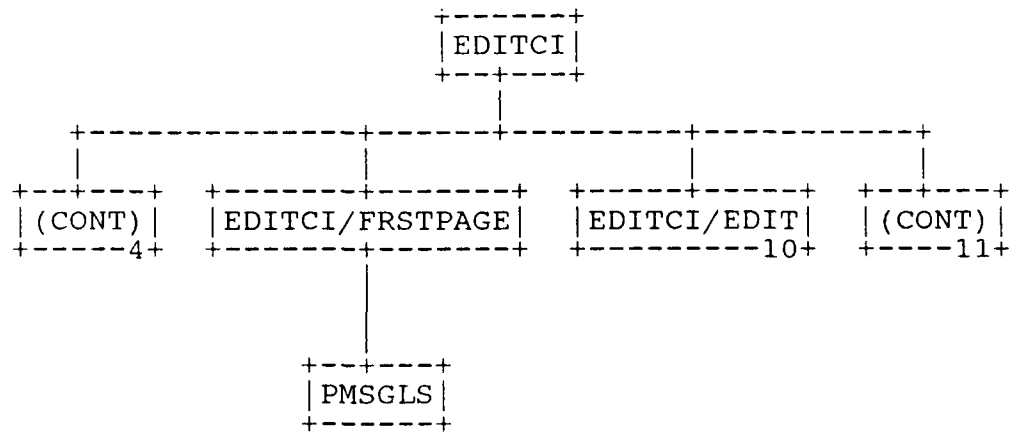
3


TE/MAIN
(CONT)
1
TERMFP
TRMNAT

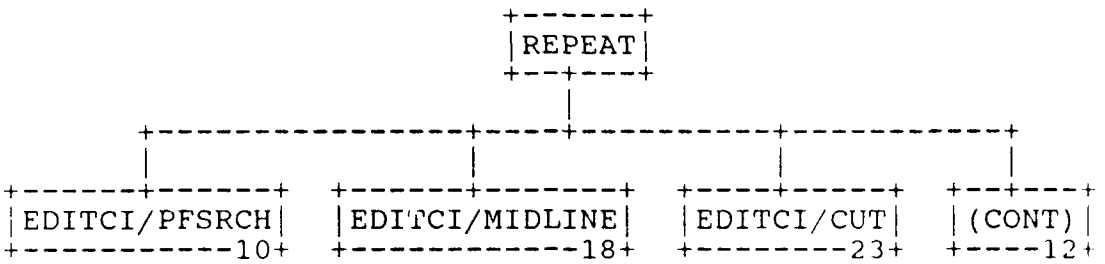
4


EDITCI
(CONT)
2
STRCPY
EDITCI/DOCMD
(CONT)
5
NAME
REPEAT
6
MARGIN
REPLACE
7
CLEAR
8
(CONT)
9
MEMCPY
ATOI

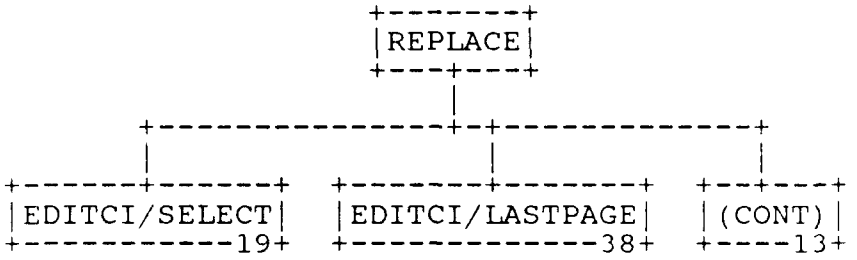
5


EDITCI
(CONT)
4
EDITCI/FRSTPAGE
EDITCI/EDIT
10
(CONT)
11
PMSGLS

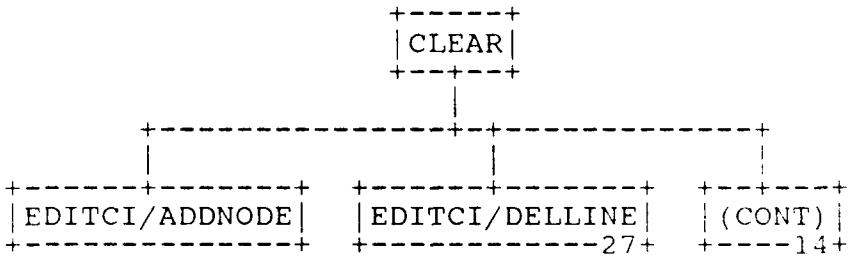
6


REPEAT
EDITCI/PFSRCH
10
EDITCI/MIDLINE
18
EDITCI/CUT
23
(CONT)
12

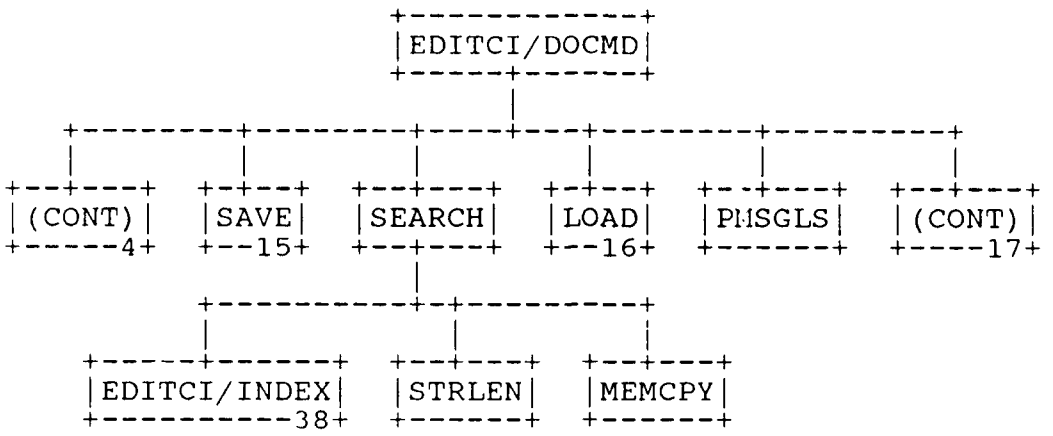
7


REPLACE
EDITCI/SELECT
19
EDITCI/LASTPAGE
38
(CONT)
13

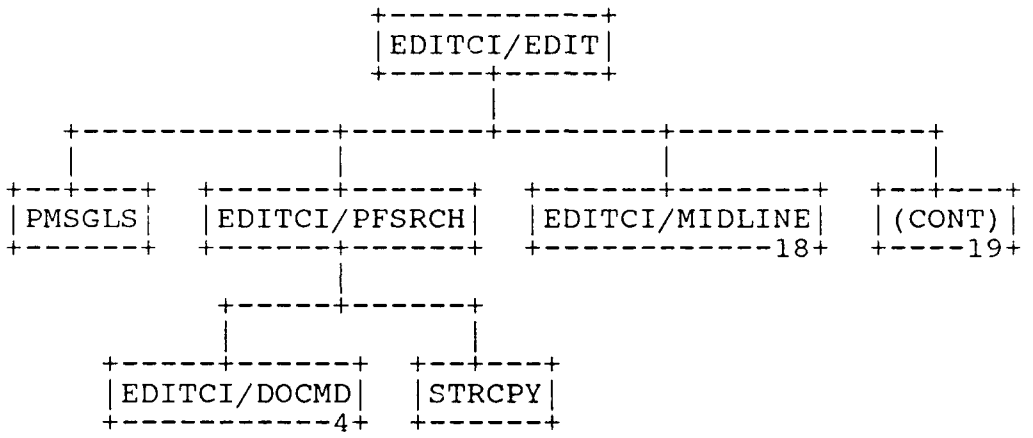
8


CLEAR
EDITCI/ADDNODE
EDITCI/DELLINE
27
(CONT)
14

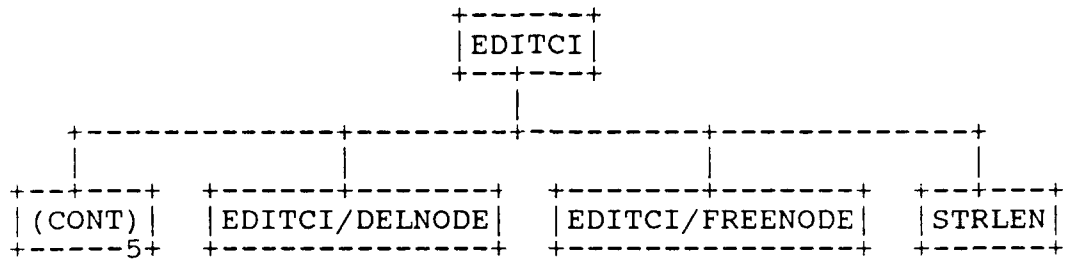
9


EDITCI/DOCMD
(CONT)
4
SAVE
15
SEARCH
LOAD
16
PMSGLS
(CONT)
17
EDITCI/INDEX
38
STRLEN
MEMCPY

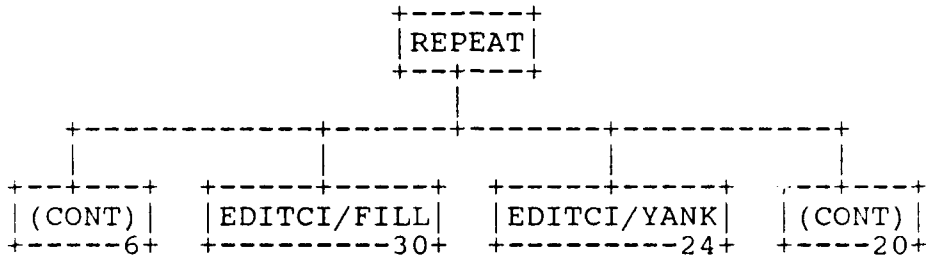
10


EDITCI/EDIT
PMSGLS
EDITCI/PFSRCH
EDITCI/MIDLINE
18
(CONT)
19
EDITCI/DOCMD
4
STRCPY

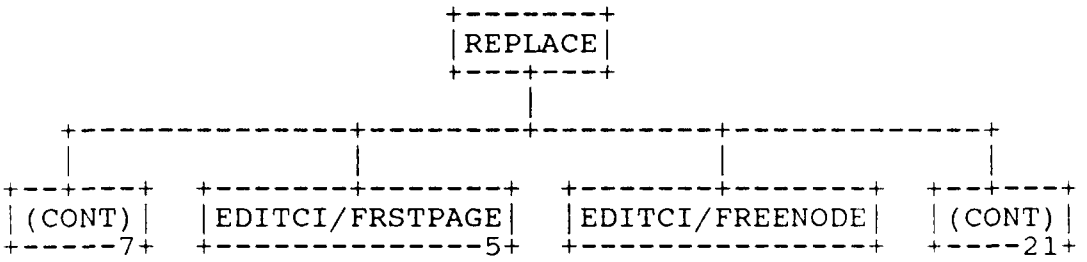
11


EDITCI
(CONT)
5
EDITCI/DELNODE
EDITCI/FREENODE
STRLEN

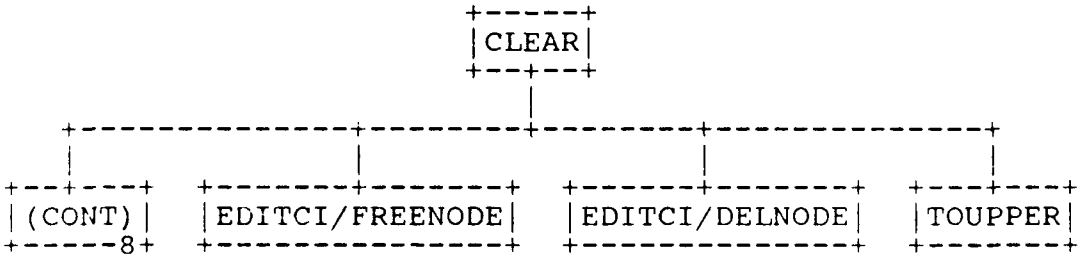
12


REPEAT
(CONT)
6
EDITCI/FILL
30
EDITCI/YANK
24
(CONT)
20

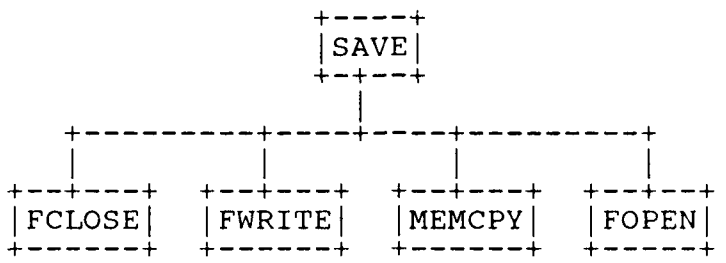
13


REPLACE
(CONT)
7
EDITCI/FRSTPAGE
5
EDITCI/FREENODE
(CONT)
21

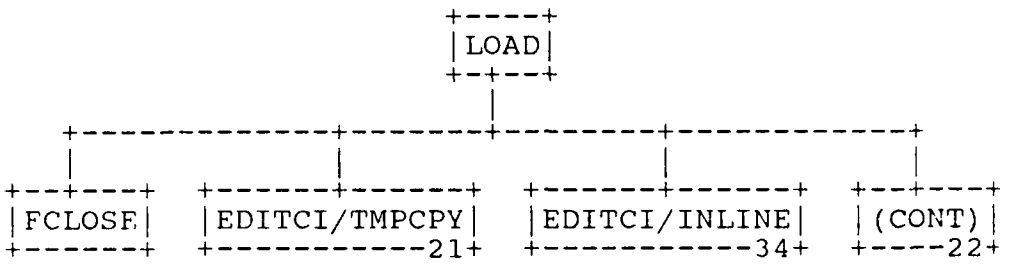
14


CLEAR
(CONT)
8
EDITCI/FREENODE
EDITCI/DELNODE
TOUPPER

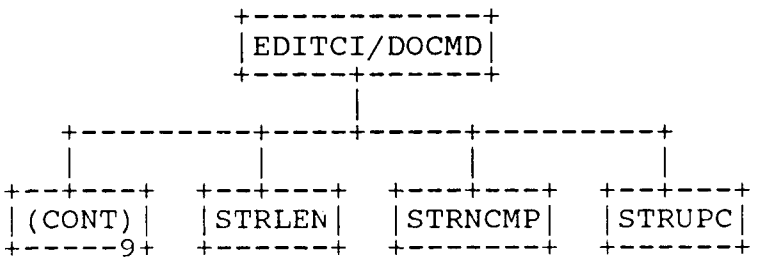
15


SAVE
FCLOSE
FWRITE
MEMCPY
FOPEN

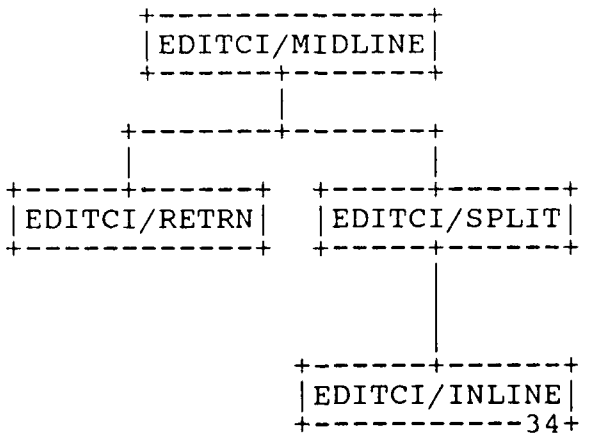
16


LOAD
FCLOSE
EDITCI/TMPCPY
21
EDITCI/INLINE
34
(CONT)
22

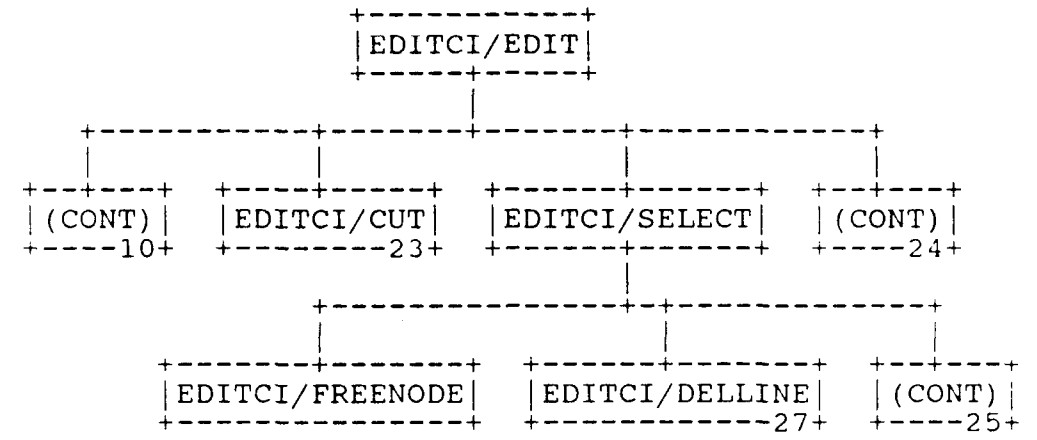
17


EDITCI/DOCMD
(CONT)
9
STRLEN
STRNCMP
STRUPC

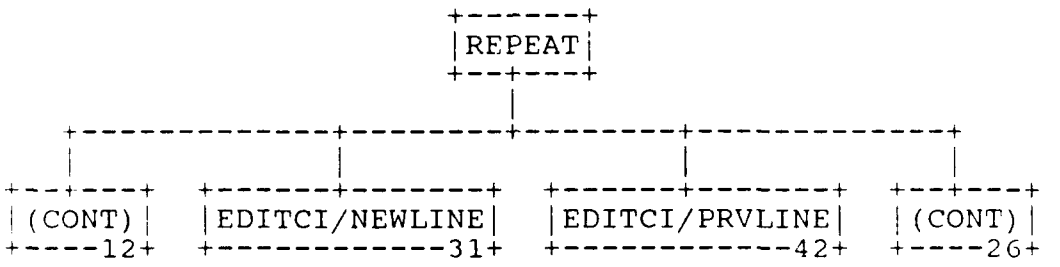
18


EDITCI/MIDLINE
EDITCI/RETRN
EDITCI/SPLIT
EDITCI/INLINE
34

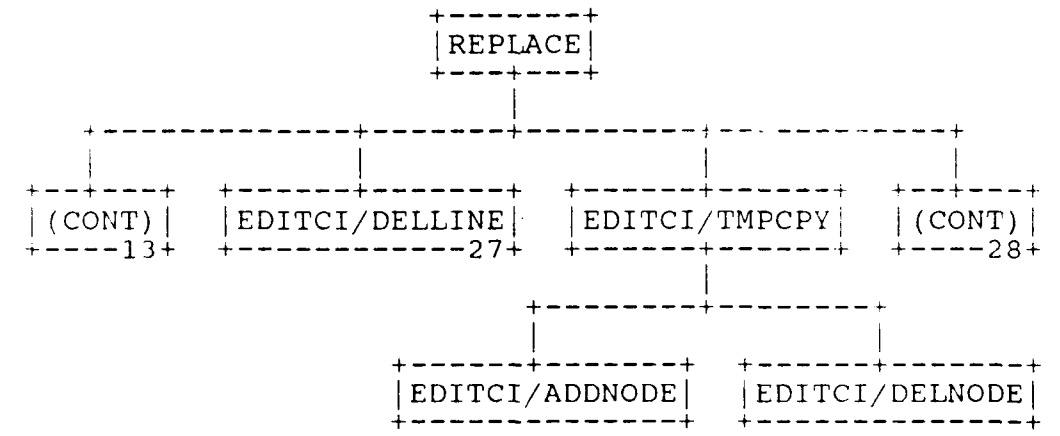
19


EDITCI/EDIT
(CONT)
10
EDITCI/CUT
23
EDITCI/SELECT
(CONT)
24
EDITCI/FREENODE
EDITCI/DELLINE
27
(CONT)
25

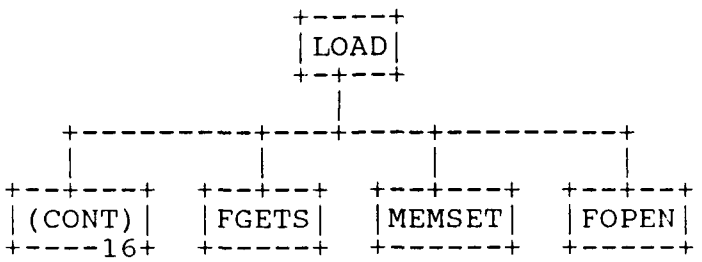
20


REPEAT
(CONT)
12
EDITCI/NEWLINE
31
EDITCI/PRVLINE
42
(CONT)
26

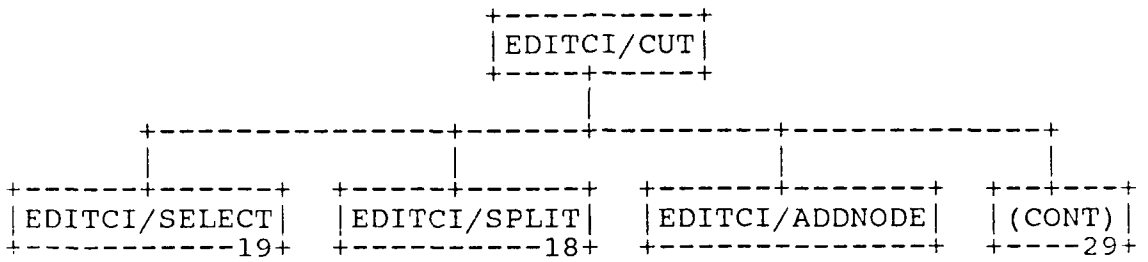
21


REPLACE
(CONT)
13
EDITCI/DELLINE
27
EDITCI/TMPCPY
(CONT)
28
EDITCI/ADDNODE
EDITCI/DELNODE

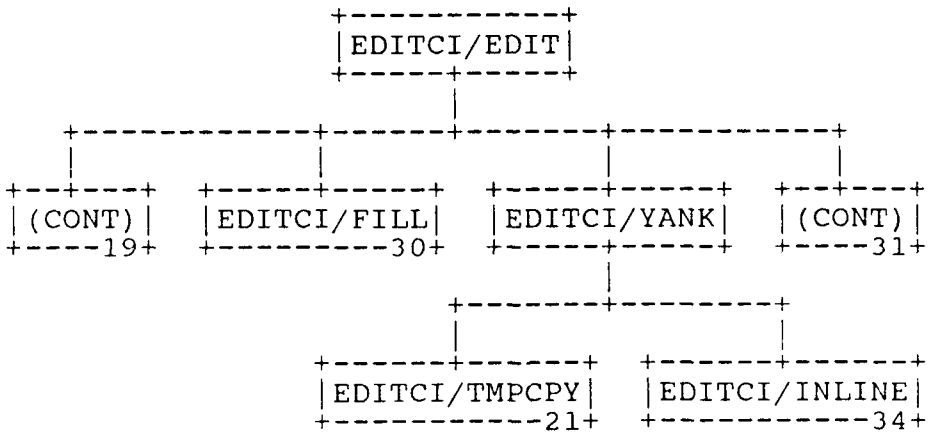
22


LOAD
(CONT)
16
FGETS
MEMSET
FOPEN

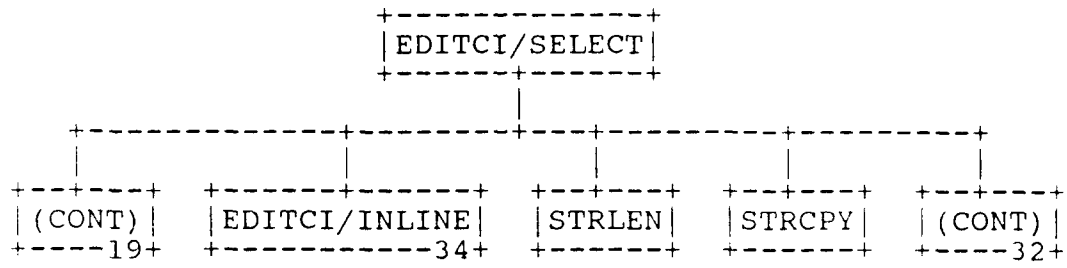
23


EDITCI/CUT
EDITCI/SELECT
19
EDITCI/SPLIT
18
EDITCI/ADDNODE
(CONT)
29

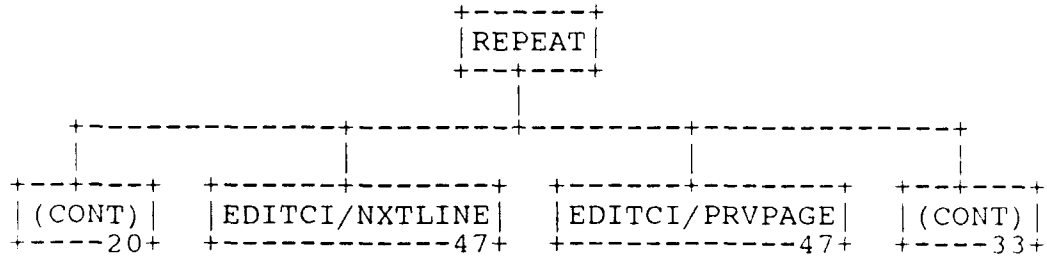
24


EDITCI/EDIT
(CONT)
19
EDITCI/FILL
30
EDITCI/YANK
(CONT)
31
EDITCI/TMPCPY
21
EDITCI/INLINE
34

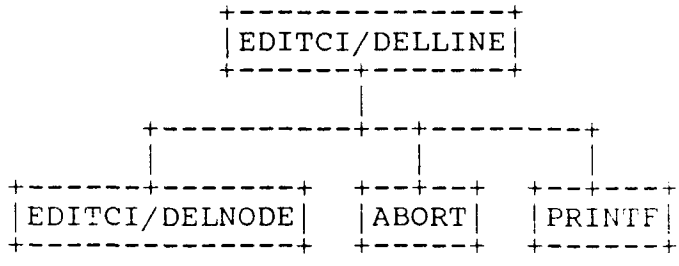
25


EDITCI/SELECT
(CONT)
19
EDITCI/INLINE
34
STRLEN
STRCPY
(CONT)
32

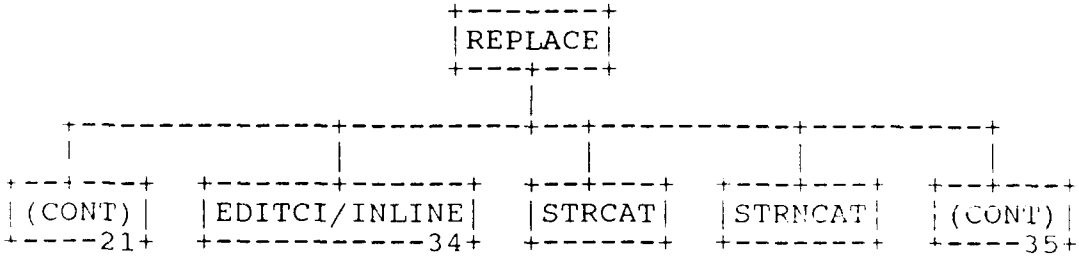
26


REPEAT
(CONT)
20
EDITCI/NXTLINE
47
EDITCI/PRVPAGE
47
(CONT)
33

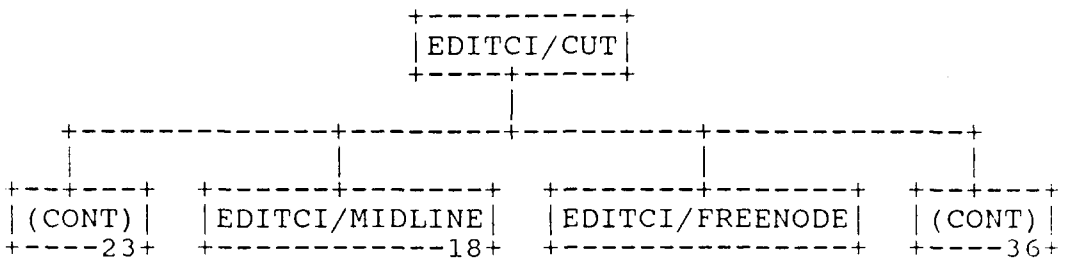
27


EDITCI/DELLINE
EDITCI/DELNODE
ABORT
PRINTF

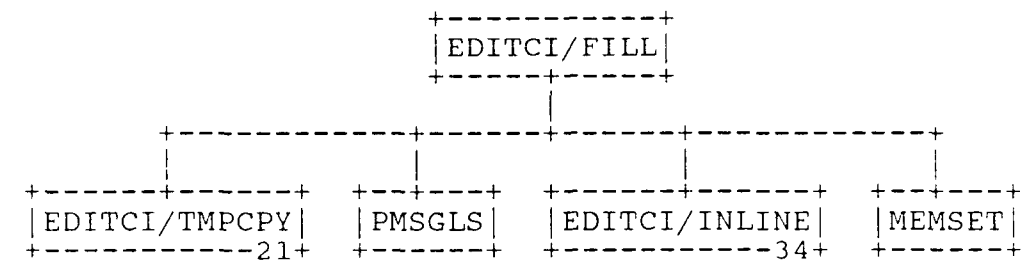
28


REPLACE
(CONT)
21
EDITCI/INLINE
34
STRCAT
STRNCAT
(CONT)
35

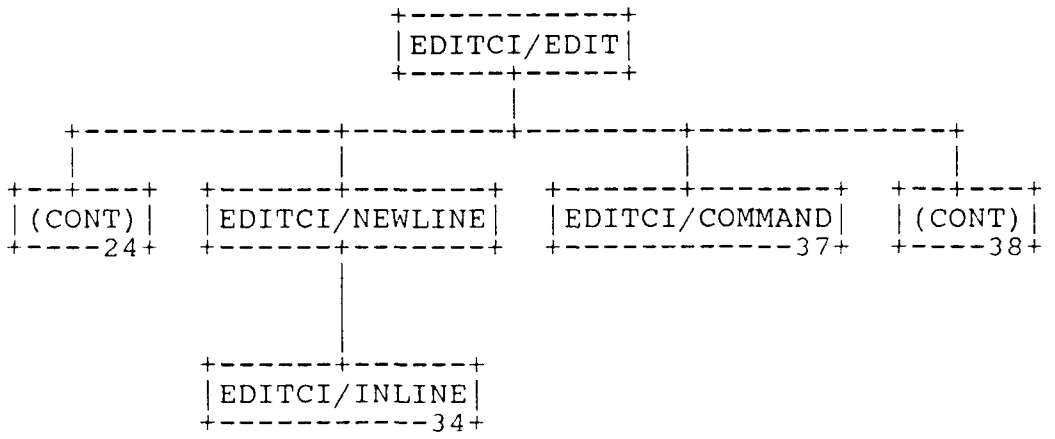
29


EDITCI/CUT
(CONT)
23
EDITCI/MIDLINE
18
EDITCI/FREENODE
(CONT)
36

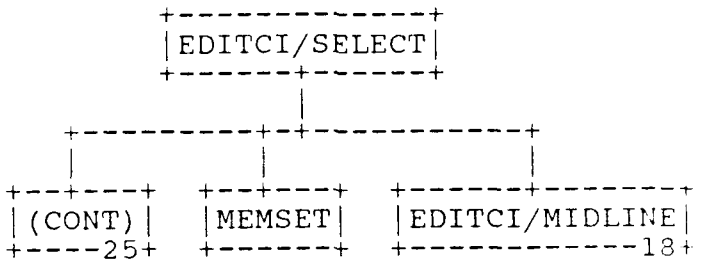
30


EDITCI/FILL
EDITCI/TMPCPY
21
PMSGLS
EDITCI/INLINE
34
MEMSET

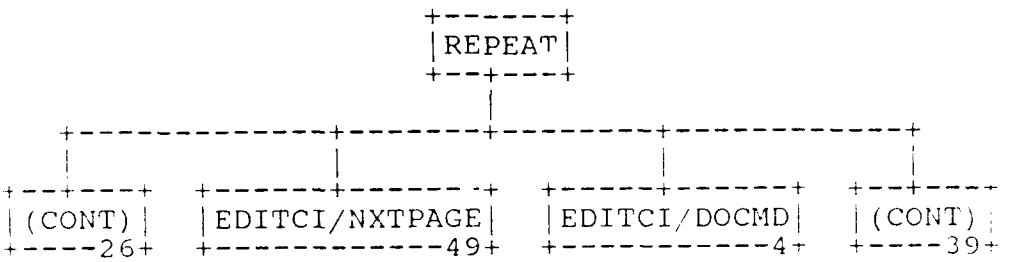
31


EDITCI/EDIT
(CONT)
24
EDITCI/NEWLINE
EDITCI/COMMAND
37
(CONT)
38
EDITCI/INLINE
34

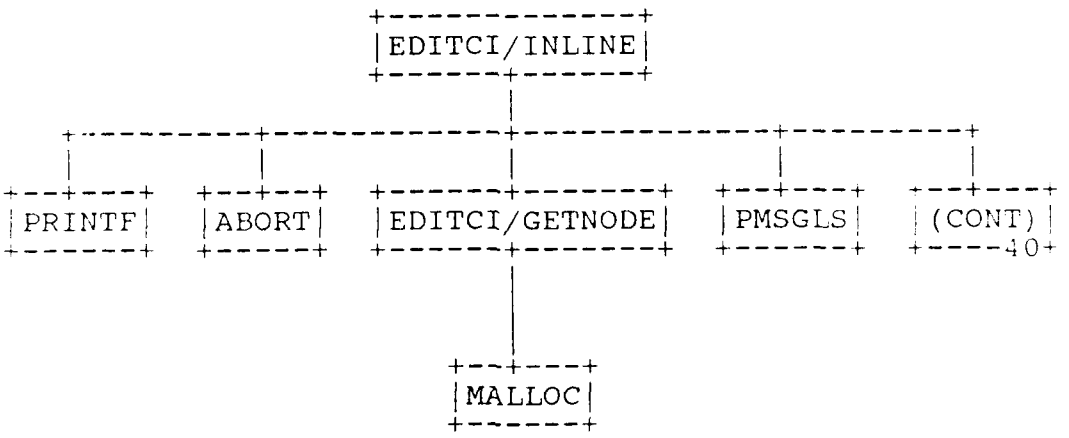
32


EDITCI/SELECT
(CONT)
25
MEMSET
EDITCI/MIDLINE
18

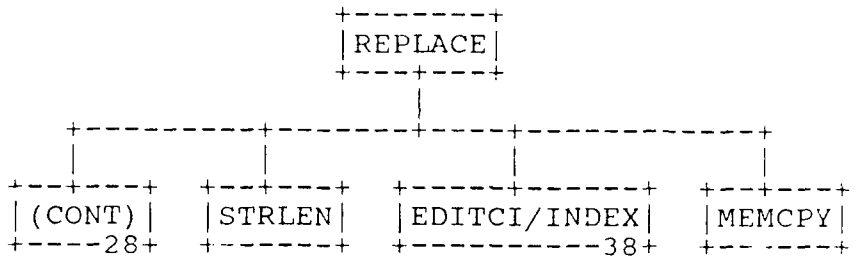
33


REPEAT
(CONT)
26
EDITCI/NXTPAGE
49
EDITCI/DOCMD
4
(CONT)
39

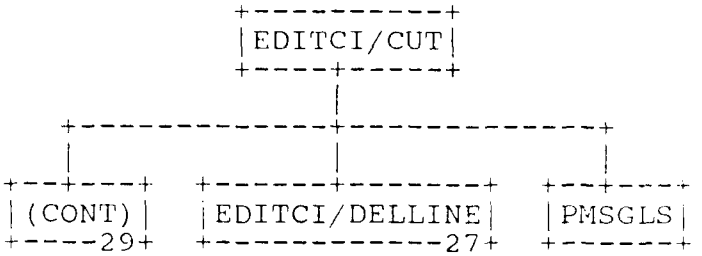
34


EDITCI/INLINE
PRINTF
ABORT
EDITCI/GETNODE
PMSGLS
(CONT)
40
MALLOC

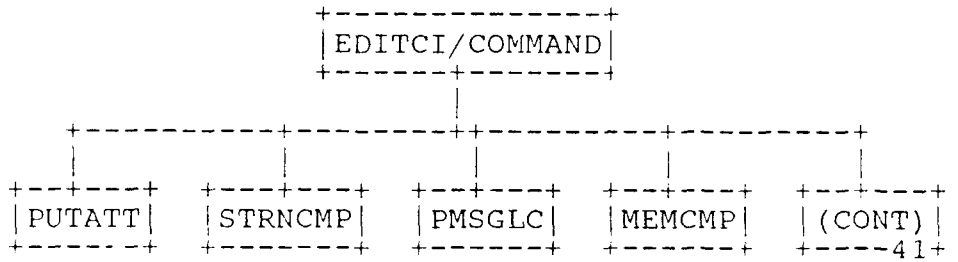
35


REPLACE
(CONT)
28
STRLEN
EDITCI/INDEX
38
MEMCPY

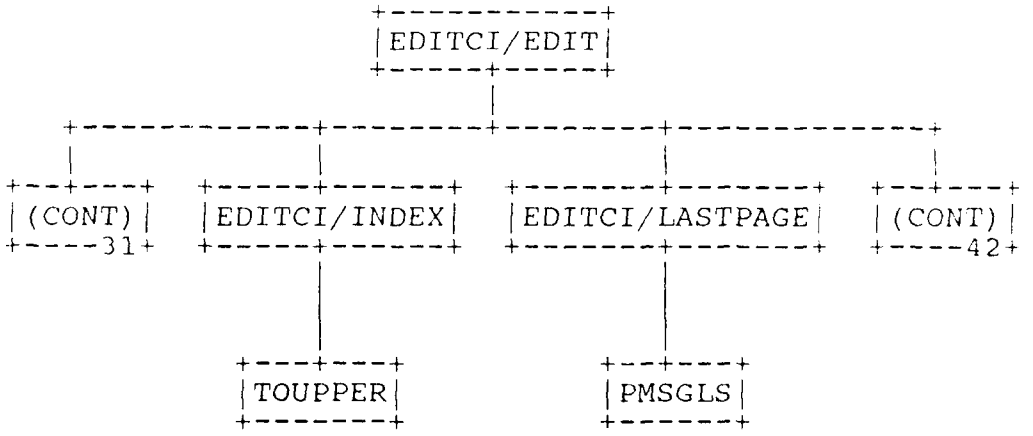
36


EDITCI/CUT
(CONT)
29
EDITCI/DELLINE
27
PMSGLS

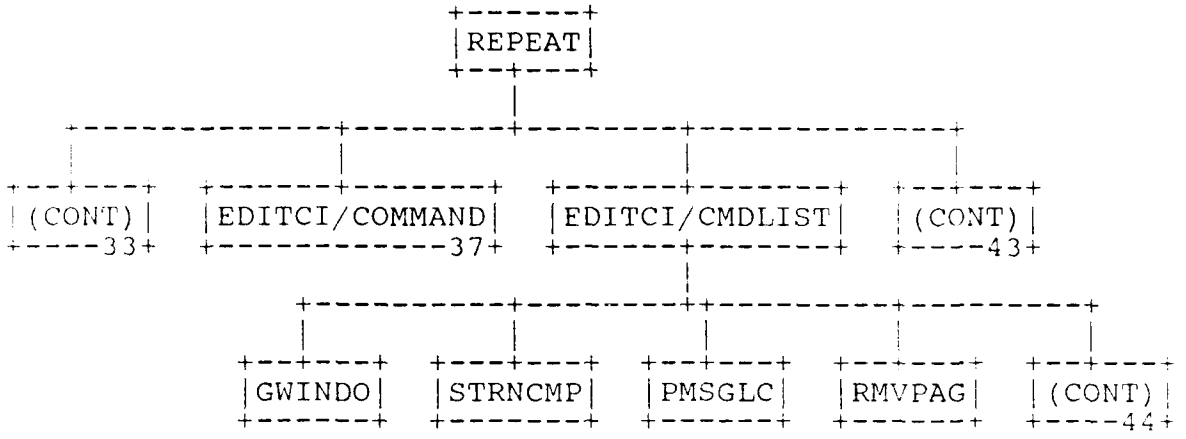
37


EDITCI/COMMAND
PUTATT
STRNCMP
PMSGLC
MEMCMP
(CONT)
41

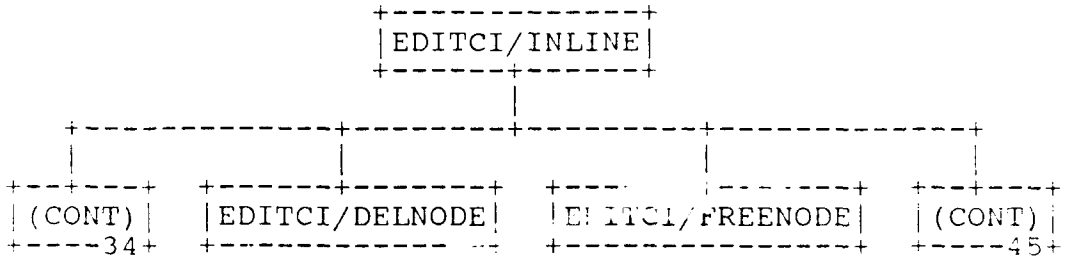
38


EDITCI/EDIT
(CONT)
31
EDITCI/INDEX
EDITCI/LASTPAGE
(CONT)
42
TOUPPER
PMSGLS

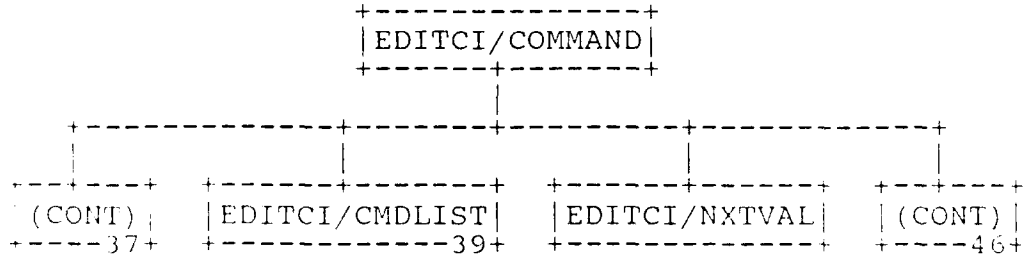
39


REPEAT
(CONT)
33
EDITCI/COMMAND
37
EDITCI/CMDLIST
(CONT)
43
GWINDO
STRNCMP
PMSGLC
RMVPAG
(CONT)
44

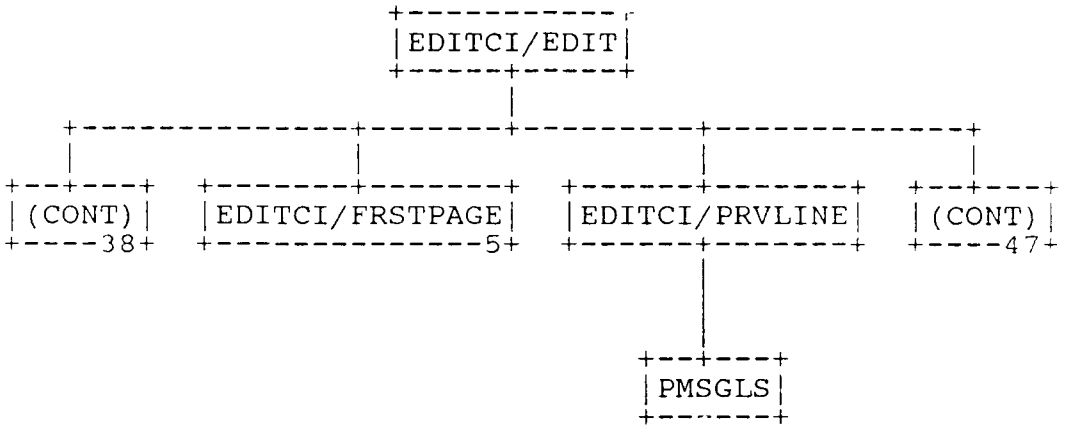
40


EDITCI/INLINE
(CONT)
34
EDITCI/DELNODE
(CONT)
45

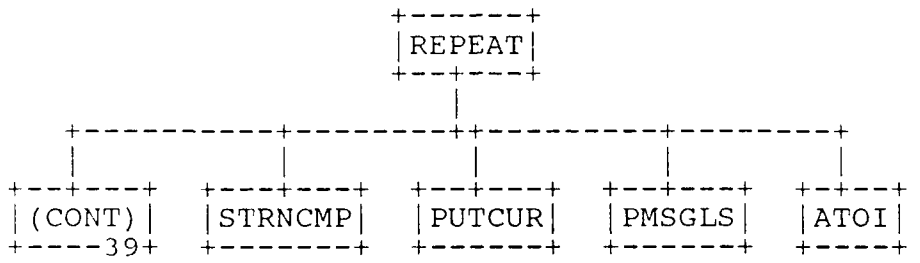
41


EDITCI/COMMAND
(CONT)
37
EDITCI/CMDLIST
39
EDITCI/NXTVAL
(CONT)
46

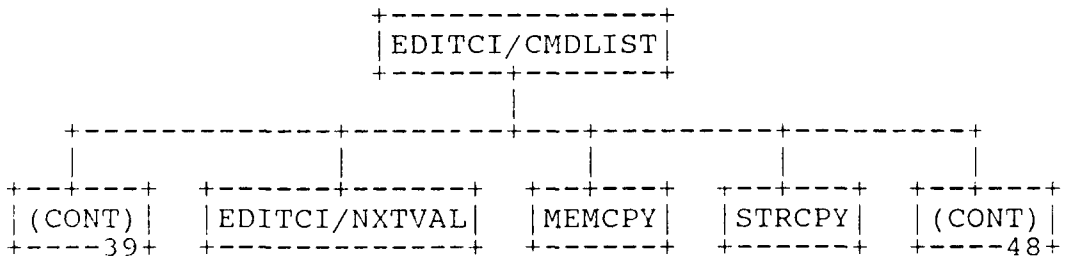
42


EDITCI/EDIT
(CONT)
38
EDITCI/FRSTPAGE
5
EDITCI/PRVLINE
(CONT)
47
PMSGLS

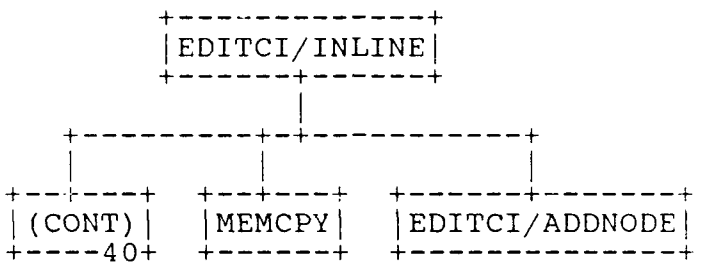
43


REPEAT
(CONT)
39
STRNCMP
PUTCUR
PMSGLS
ATOI

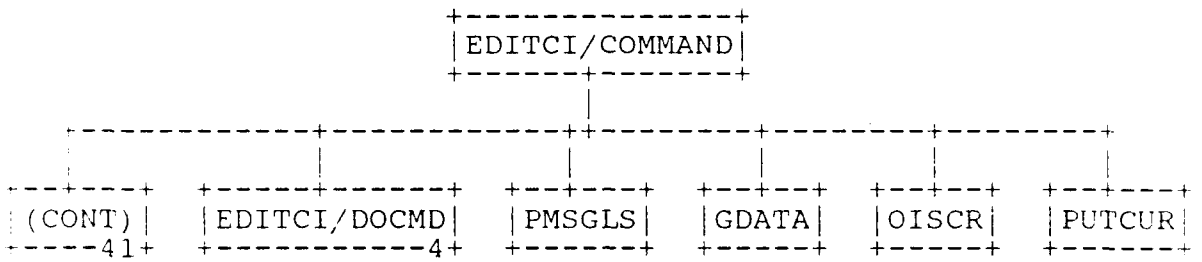
44


EDITCI/CMDLIST
(CONT)
39
EDITCI/NXTVAL
MEMCPY
STRCPY
(CONT)
48

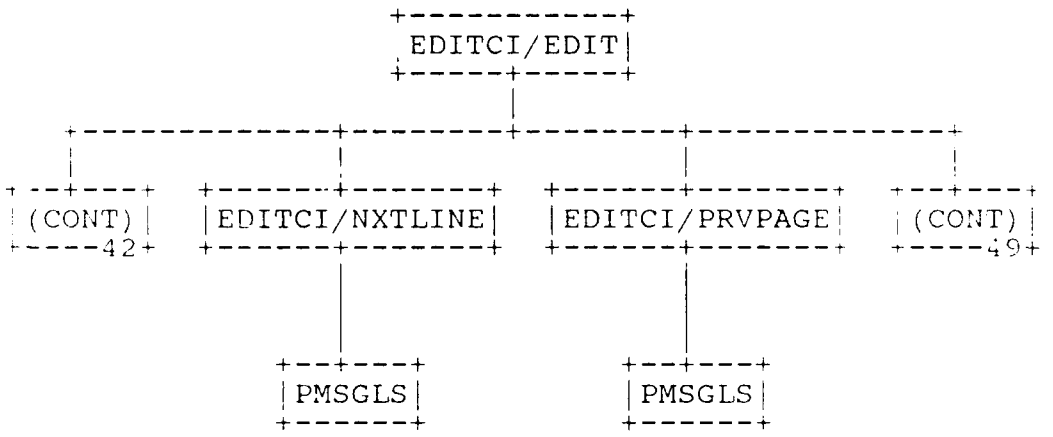
45


EDITCI/INLINE
(CONT)
40
MEMCPY
EDITCI/ADDNODE

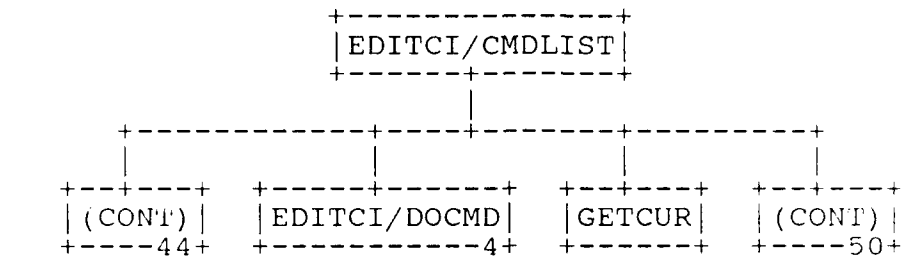
46


EDITCI/COMMAND
(CONT)
41
EDITCI/DOCMD
4
PMSGLS
GDATA
OISCR
PUTCUR

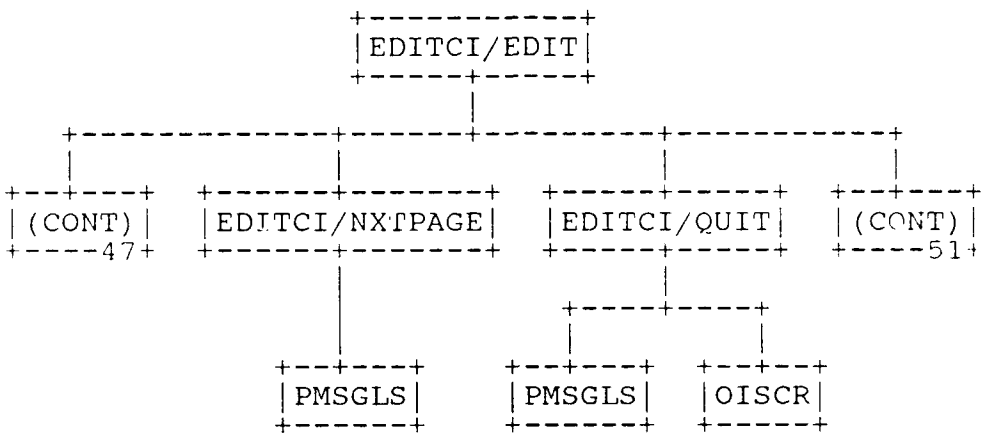
47


EDITCI/EDIT
(CONT)
42
EDITCI/NXTLINE
EDITCI/PRVPAGE
(CONT)
49
PMSGLS
PMSGLS

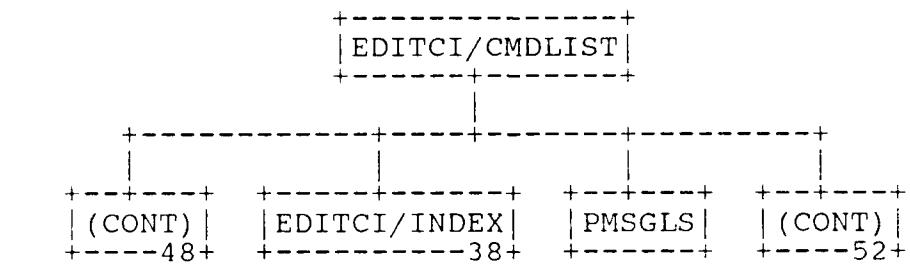
48


EDITCI/CMDLIST
(CONT)
44
EDITCI/DOCMD
4
GETCUR
(CONT)
50

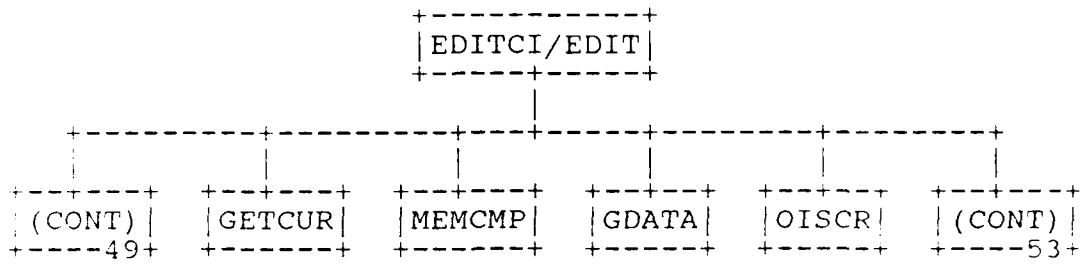
49


EDITCI/EDIT
(CONT)
47
EDITCI/NXTPAGE
EDITCI/QUIT
(CONT)
51
PMSGLS
PMSGLS
OISCR

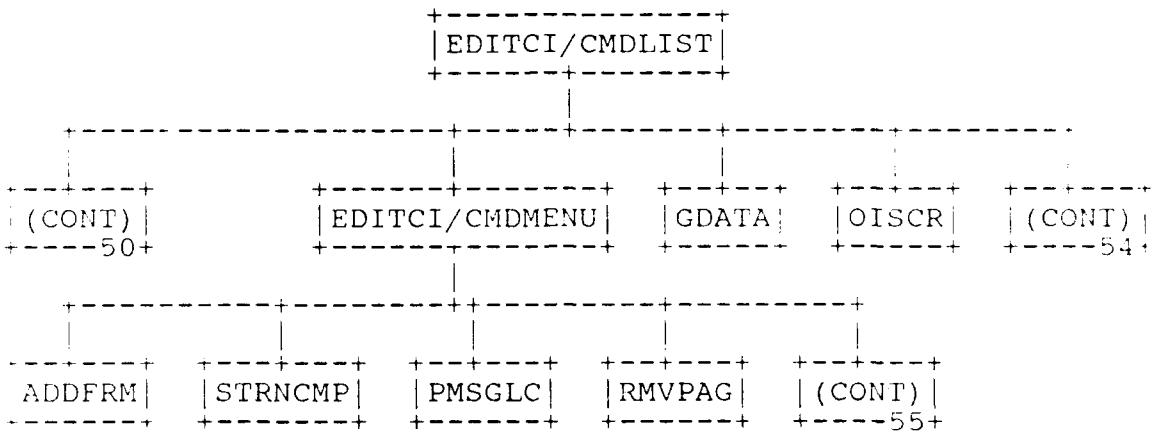
50


EDITCI/CMDLIST
(CONT)
48
EDITCI/INDEX
38
PMSGLS
(CONT)
52

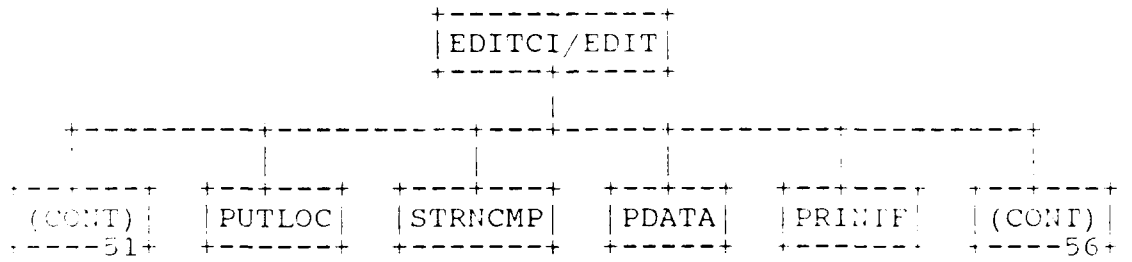
51


EDITCI/EDIT
(CONT)
49
GETCUR
MEMCMP
GDATA
OISCR
(CONT)
53

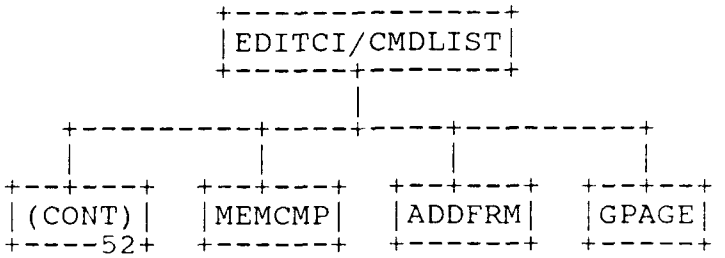
52


EDITCI/CMDLIST
(CONT)
50
EDITCI/CMDMENU
GDATA
OISCR
(CONT)
54
ADDFRM
STRNCMP
PMSGLC
RMVPAG
(CONT)
55

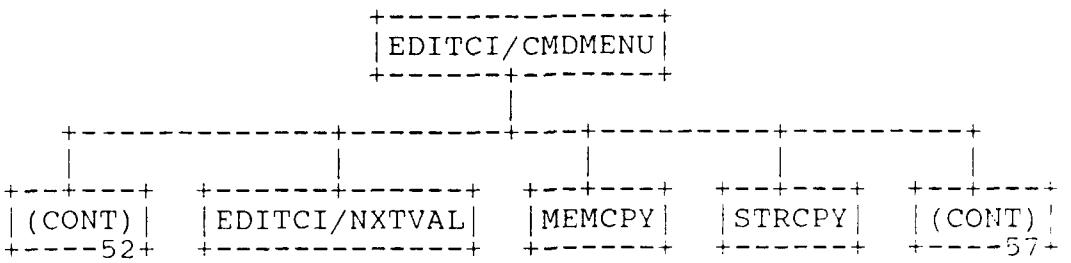
53


EDITCI/EDIT
(CONT)
51
PUTLOC
STRNCMP
PDATA
PRINTF
(CONT)
56

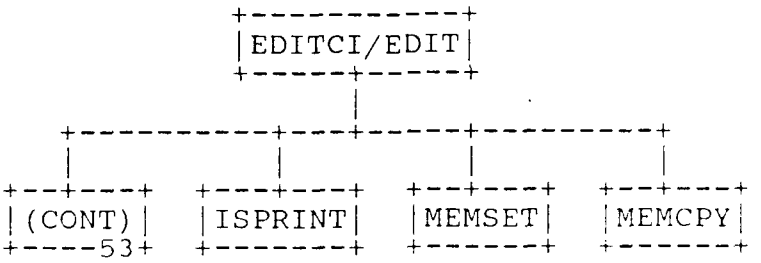
54

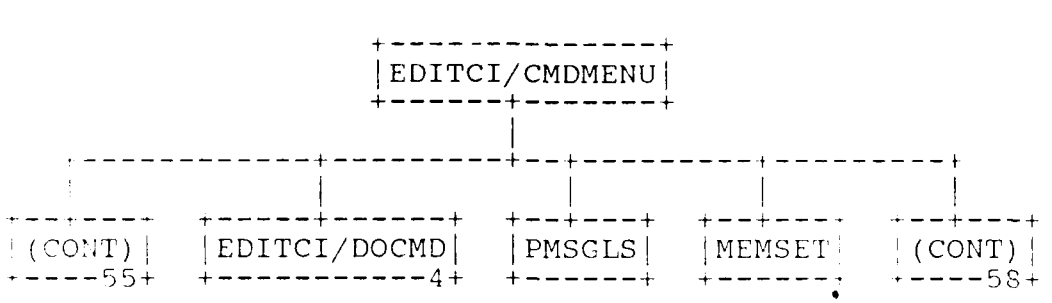

EDITCI/CMDLIST
(CONT)
52
MEMCMP
ADDFRM
GPAGE

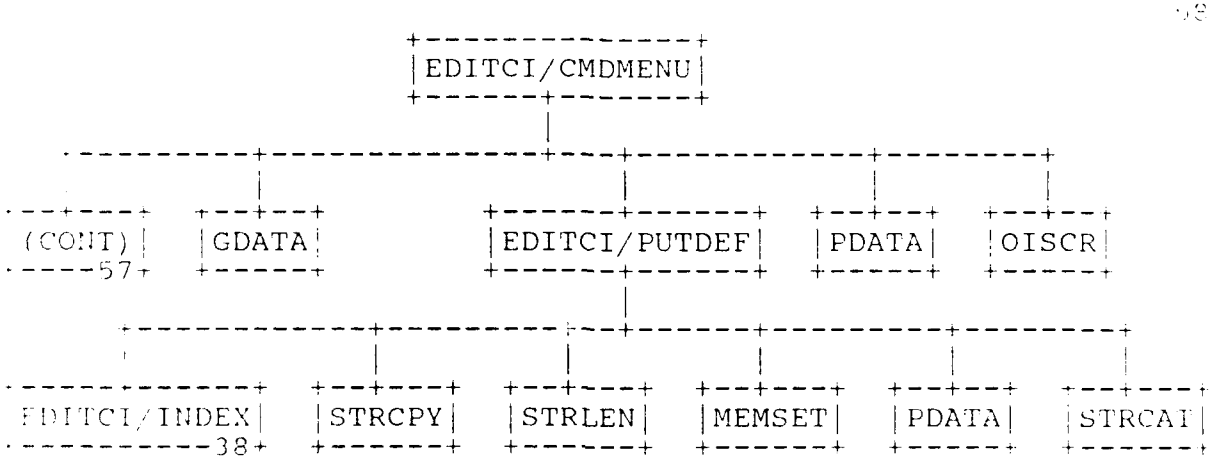
55


EDITCI/CMDMENU
(CONT)
52
EDITCI/NXTVAL
MEMCPY
STRCPY
(CONT)
57

56


EDITCI/EDIT
(CONT)
53
ISPRINT
MEMSET
MEMCPY


57
EDITCI/CMDMENU
(CONT)
55
EDITCI/DOCMD
4
PMSGLS
MEMSET
(CONT)
58


EDITCI/CMDMENU
(CONT)
57
GDATA
EDITCI/PUTDEF
PDATA
OISCR
EDITCI/INDEX
38
STRCPY
STRLEN
MEMSET
PDATA
STRCAT

ABORT
ADDFRM
ATOI
CLEAR.....8
EDITCI.....2
EDITCI/ADDNODE
EDITCI/CMDLIST....39
EDITCI/CMDMENU....52
EDITCI/COMMAND....37
EDITCI/CUT.....23
EDITCI/DELLINE....27
EDITCI/DELNODE
EDITCI/DOCMD.....4
EDITCI/EDIT.....10
EDITCI/FILL.....30
EDITCI/FREENODE
EDITCI/FRSTPAGE....5
EDITCI/GETNODE....34
EDITCI/INDEX.....38
EDITCI/INLINE.....34
EDITCI/LASTPAGE...38
EDITCI/MIDLINE....18
EDITCI/NEWLINE....31
EDITCI/NXTLINE....47
EDITCI/NXTPAGE....49
EDITCI/NXTVAL
EDITCI/PFSRCH.....10
EDITCI/PRVLINE....42
EDITCI/PRVPAGE....47
EDITCI/PUTDEF.....58
EDITCI/QUIT.....49
EDITCI/RETRN
EDITCI/SELECT.....19
EDITCI/SPLIT.....18
EDITCI/TMPCPY.....21
EDITCI/YANK.....24
ESCPY
FCLOSE
FGETS
FOPEN
FWRITE
GDATA
GETCUR
GETMORE
GPAGE
GWINDO
INITAL
INITFP
ISPRINT
LOAD.....16
MALLOC
MARGIN.....4
MEMCMP
MEMCPY
MEMSET
NAME.....4
OISCR
PDATA
PMSGLC
PMSGLS
PRINTF
PUTATT
PUTCUR
PUTLOC
REPEAT.....6
REPLACE.....7
RMVPAG
SAVE.....15
SEARCH.....9
STRCAT
STRCPY
STRLEN
STRNCAT
STRNCMP
STRUPC
TE/MAIN.....1
TERMFP
TOUPPER
TRMNAT

3.11 Program Listings Comments

This information is contained in the Module Descriptions in section 3.10.

SECTION 4

QUALITY ASSURANCE PROVISIONS

4.1 Introduction and Definitions

"Testing" is a systematic process that may be preplanned and explicitly stated. Test techniques and procedures may be defined in advance, and a sequence of test steps may be specified. "Debugging" is the process of isolation and correction of the cause of an error.

"Antibugging" is defined as the philosophy of writing programs in such a way as to make bugs less likely to occur and when they do occur, to make them more noticeable to the programmer and the user. In other words, as much error checking as is practical and possible in each routine should be performed.

4.2 Computer Programming Test and Evaluation

The quality assurance provisions for test consists of the normal testing techniques that are accomplished during the construction process. They consist of design and code walk-throughs, unit testing, and integration testing. These tests are performed by the design team. Structured design, design walk-through and the incorporation of "antibugging" facilitate this testing by exposing and addressing problem areas before they become coded "bugs."

☆U.S. Government Printing Office:1992—648-127/62245